Database Extensions, Page Fragments and Plugins

Released in PowerSchool 7.9

Key Documents on PowerSource:

“Database Extensions Visual Walkthrough” - ID: 70768
https://powersource.pearsonschoolsystems.com/article/70768

“Advanced User Guide for Database Extensions” - ID: 70667
https://powersource.pearsonschoolsystems.com/article/70667

Errors:

- Page 16 (Numbered as Page 13) at the top of the page:
  - it reads “/admin/some.directory”
  - it should be “/admin/some_directory”
- Same page, in the graphic about ⅔ of the way down where it shows an example as:
  “home.html.Emergency_Numbers.content.footer.txt”
  It should be:
  “home.Emergency_Numbers.content.footer.txt”

Overview

- Database Extensions is the name for the new method for custom data entry points in PowerSchool.
- It will currently run side-by-side with the old way of doing custom data, but eventually will completely replace it.
- **Before:**
  - Users go to System - Custom Fields/Screens
  - Only Student, Staff (teachers table), Course and Section fields were available to be created this way
  - Only text fields could be created
  - All data went into one giant custom field that had to be parsed
  - The only way to do one-to-many custom fields was to use virtual tables which is little known and very difficult to work with
- **After:**
  - Users go to System - Page and Data Management - Manage Database Extensions
  - Users create REAL tables and fields of various data types.
  - Options to migrate core custom fields and user custom fields. Currently core fields can be migrated at once, user fields can be migrated one at a time (available where you create an extended field).
  - Many, many more tables can be extended, users can create one-to-many tables and even independent tables.
Inherent grouping of fields for logical organization.
- No more 999 limit! (There is still a 999 limit per “group”, but if you have a group with more than 999 fields...... well....)
- All tables get some tracking fields: who created, when created, who modified, when modified. It won’t tell you what changed, but who and when.
- One-to-Many and Independent tables can be export from and imported into with the Data Export and Import Managers.

Core Fields Migration
- 1st 59 student core custom fields are moved to a new table called StudentCoreFields
  - ACT_Composite/Date/English/Math/Reading/Science, SAT
  - Emerg_1/2/3_Ptype/Rel
  - Guardian, Guardian_LN, Guardian_FN, Guardian_MN
  - Mo/Father_Home_Phone, Mo/Father_Day_Phone, Mo/Father_Employer
  - PrimaryLanguage, SecondaryLanguage
  - PrevStudentID, Family_Rep, Area, Dentist_Name, etc
- 2 teacher/user core custom fields are moved to a new table called UsersCoreFields
  - DOB
  - Gender

Core Fields 2 Migration
- 52 more student custom fields are migrated to the StudentCoreFields table
  - Autosend fields
  - CRT_ fields
  - EC_ fields
  - IPT_ fields
  - a few others.
- 1 course field is moved to CourseCoreFields table
  - Alt_Course_Number

User Custom Fields migration - currently only one-at-a-time. You can migrate an existing custom field from the screen where you create a new extended field.

In Version 7.10 activities are migrated to a new table called “Activities”. There are some current issues with this migration that do not harm the user experience.

Migrated fields continue to operate the “old” way without changing any existing custom pages, exports, reports, etc. PowerSchool is programmed to act like migrated fields still exist in the “old” location.

Please note the issues with migration of fields that contain more than 4000 characters.

Remember the flow:
Core Table > Extension Group(s) > Extension Table(s) > Extension Field(s)
Cautionary Notes:

- You cannot delete, change or rename extensions once created. So plan carefully and think about your users and customizations when naming them.
- For each extension group’s core table, you can only have one 1:1 extended table, but you can have multiple 1:many tables. You MUST define a 1:1 extended table first before creating any 1:many tables.
- When a plugin containing a database extension is installed or modified, you must restart PowerSchool/PowerTeacher and ReportWorks services.
- When you disable a plugin, it won’t be served from CPM, but there is no way to tell in CPM if a custom page is associated with a plugin.
- ReportWorks will show extended fields, but must be restarted first.
- It’s not obvious, but you can have more than one core table per extension group… just select the core table and then pick an existing extension group in Step 2.

Using One-to-One Extensions

- Searching, exports, list students (wherever the PS application itself asks you for a “fieldname”)
  - ExtensionGroupName.Field_Name.
  - Example: U_Students_Extension.DistrictID
- Reports that use DATs (wherever you normally put ~(fieldname))
  - ~(ExtensionGroupName.Field_Name)
  - Example: U_Students.Extension.DistrictID
- HTML pages
  - [PrimaryTable.ExtensionsGroupName]FieldName
  - Example: [Students.U_Students_Extension]DistrictID
- SQL Queries.
  - The extended database tables link on {CORETABLE}DCID
    SELECT * FROM CoreTable
    LEFT JOIN ExtendedTable ON CoreTable.DCID = ExtendedTable.StudentsDCID
  - EXAMPLE:
    SELECT s.lastfirst, s.grade_level, s2.districtid
    FROM students s
    LEFT JOIN U_DEF_EXT_STUDENTS s2 ON s.dcid = s2.studentsdcid
    WHERE s.enroll_status=0 AND s.grade_level=12 AND s.schoolid = 100
One-to-Many Extension Tables
A one-to-many extended table allows you to have multiple records that are tied back to a single parent record. For example, multiple college applications for a single student.

- To work with one-to-many tables you use the tlist_child tag. This tag will create a table, so use it outside of any other tables, but still within a standard type page that has a form tag with the usual submit button and associated hidden inputs.

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

Example:

```~[tlist_child:Students.U_CollegeApp.U_Applications;displaycols:Institution,Request_Date,Status;fieldNames:Institution,Request Date,Status;type:html]~```

- SQL query example for a one-to-many table.

```SELECT    s.lastfirst,    app.institution,    app.request_date,    app.status    FROM    students s    LEFT JOIN U_Applications app ON s.dcid = app.studentsdcid    WHERE    s.id = 2```  

Independent (Standalone) Extension Tables
Creates a table that is not associated with any existing PowerSchool table. An example might be a list of college institutions.

- To work with independent tables use the tlist_standalone tag. It works much like the tlist_child tag.

```~[tlist_standalone:<ExtensionGroup>.<ExtensionTable>;displaycols:<List of Fields>;fieldNames:<List of Column Headers>;type:<FormatName>]~```

Example:

```~[tlist_standalone:U_CollegeApp.U_Institutions;displaycols:Institution_Name,Phone,URL;fieldNames:Institution Name,Phone Number,Web Address;type:html]~```

- Example SQL Query for Independent table:

```SELECT * FROM U_Institutions```
Even more advanced?

- Refer to the Advanced User Guide for Database Extensions for special formatting of \texttt{tlist\_child} and \texttt{tlist\_standalone} tags.
- The forums are starting to discuss even more advanced applications of extended tables for when \texttt{tlist\_child} and \texttt{tlist\_standalone} aren't sufficient.

Page Fragments

Page fragments are ways to customize any existing PowerSchool page or even new pages without customizing the page itself. Rather, you create a snippet of code that is inserted dynamically into the existing page via an “insertion point”, a special location in the source code of a page.

- With insertion points, \textit{the original source page does not have to be customized} in order to add new content to that page. This can help dramatically cut down on the number of custom pages that need to be created and subsequently updated when a new version of PowerSchool is released.
- You can physically move fragments around on the page using client-side DOM manipulation via standardized metadata.
- You can insert a page fragment into a wildcard, thus being able to customize multiple pages at once - wherever that wildcard is used.

A page fragment is simply a snippet of content to be added to a target page. It could be something simple like the following example:

\begin{verbatim}
<p>Hello world! I'm an auto-inserted page fragment.</p>
\end{verbatim}

Or, a page fragment could be a complex combination of HTML code and jQuery scripts. Because page fragments will be inserted into existing PowerSchool HTML pages they do not require any of the standards HTML \texttt{<head>}, \texttt{<body>}, or other tags. The main page already contains those tags.

\textbf{The naming of your page fragment is the key.} Always save your page fragments to the same directory where the source page or wildcard exists.

\begin{itemize}
  \item Name\_of\_file (without the “.html”) +
  \item .name\_for\_page\_fragment (whatever you want to name it) +
  \item .insertion\_point (a common insertion point is “content.footer”) +
  \item .txt
\end{itemize}

Example: \texttt{home.Emergency\_Numbers.content.footer.txt}
Database Extension Plugins

PowerSchool provides an exporting function that builds a complete plugin package containing database extensions and custom pages/fragments in one file. This plugin can then be easily imported into other PowerSchool servers in one simple step.

You work with Plugins from System - System Settings - Plugin Management Dashboard

For more on how to create/export plugins and other information, please refer to the “Advanced User Guide for Database Extensions”.

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

Plugin advantages:

- An improvement over CPM “Import” ability
- Easy to share
- Can import files that CPM cannot import, like image and PDF files.
- Complete “package” of database extensions plus pages/fragments. Files will show up in CPM.
- Can be disabled with one click. All the files will still be there and even show in CPM, but will not be served.
- The plugin can be deleted which deletes the files/pages. However, extended tables and fields CANNOT be deleted.