BECK PNEZD Revit Add-In

The purpose of this Add-In is to manage point elements within the Revit model. This utility contains functions to import, manage data, and export reports of point families to Microsoft Excel 2010, TXT (tab delimited), and CSV (comma separated value) formats.

# Installation from CASE Server

The Add-In can be installed by pasting the link into the address bar of Internet Explorer:   
[www.case-dev.com/revit/installer$/beck/2012/BECK.PointManagement/BECK.PointManagement.Updater.application](http://www.case-dev.com/revit/installer$/beck/2012/BECK.PointManagement/BECK.PointManagement.Updater.application)

The installer utilizes Microsoft’s ClickOnce installation technology and does not require administrative privileges to install. The files install into the user’s profile but can be manually copied into the machine’s installation folder on other machines as necessary for distribution.

# Distribution Internal to BECK

The .addin can be freely distributed internal of BECK by copying the required files into one of the .addin locations described below:

**User Profile Add-Ins Location**

%USERPROFILE%\AppData\Roaming\Autodesk\Revit\Addins\2012

**Machine Wide Add-Ins Location**

C:\ProgramData\Autodesk\Revit\Addins\2012

## Required Installation Files

* BECK.PointManagement.Documentation.docx
* BECK.PointManagement.addin
* BECK.PointManagement.dll
* BECK\_PointsManager\_Template.xlsx

# File Based Template

An import/export file template has been provided in Microsoft Excel 2010 format named “BECK\_PointsManager\_Template.xlsx”. The TXT and CSV file types follow a similar template format but have not been provided as the tool will generate them automatically on export.

## The Microsoft Excel 2010 Template

There are a few basic requirements that the Excel template needs in order to function properly. The workbook name must be named as the category that the data on it contains. You can manage multiple categories of elements, but since the available parameters between categories can change, so must the worksheet that manages them.

The first row of the spreadsheet must contain the parameter names and data headers. The first column must be named “ElementID” and will manage the ElementID property for each point element that the row represents. The naming format for the PNEZD data is so that the parameter name for that data node is contained within the brackets [].

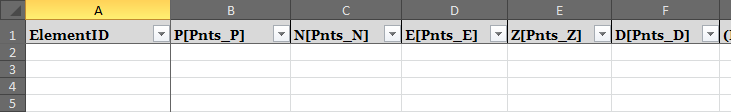


Figure 1 – Sample Excel PNEZD and ElementID Headers

In the example worksheet shown above, the parameter named for the “P” data node would be “Pnts\_P.” This parameter can be changed by editing the name of the parameter within the brackets. An error will be thrown if a duplicated “P” entry is identified within the same category.

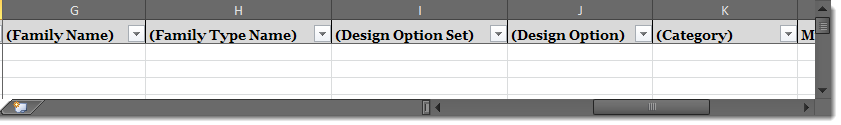


Figure 2 – Sample Supported Properties Headers

There are 5 supported element properties that when entered as required fields along the header row will be included on export only. These data fields will not sync back into the model and are read-only. The image above shows how these properties need to be entered along header row. These are entirely optional and are as follows:

* (Family Name)
* (Family Type Name)
* (Design Option Set)
* (Design Option)
* (Category)

Aside from the above mentioned properties and parameters, it is possible to manage up to five additional parameters as well. Managing additional parameters is done by simply listing the name of the parameter within the header row. If the specified parameter is found within the category, the data for this parameter will be synced in and out of this worksheet matching that parameter name. These added parameter names should not be enclosed in brackets or parentheses.

## The CSV and TXT File Format Requirements

The CSV and TXT file formats require that the file be named prefixed with the category name followed by an underscore (\_). This tells the importer what category that the elements belong to. The same element properties and parameter naming outlined above in the Excel template are the same for these formats.

# The 6 Ribbon Commands

All of the included functionality can be accessed from the six command buttons found on the custom “BECK” ribbon tab as show in the image below.

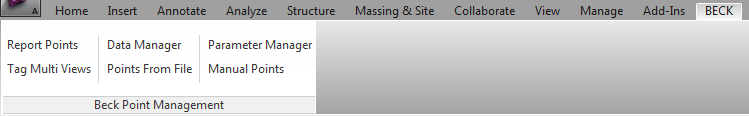


Figure 3 - The Custom BECK Ribbon Tab

## Report Points

This command button is used to access the point reporting features. The file formats available as export from the reporting feature includes Microsoft Excel 2010, CSV, and Tab Delimited TXT file formats. The Excel export format relies on the existence of a template file adjacent to the DLL file (File Based Template mentioned above).

### Coordinate Systems and Filtering

The tool supports the export of coordinates in Project, Shared, or Named systems. At the very minimum, the user must filter on category or family in order to report point data. The user can narrow down the export scope by Family & Type, Category, Level, and Design Option. Any combination of filtering can be applied for an export. Click the Export Points button to be prompted to select your exported file name and location and export the data.

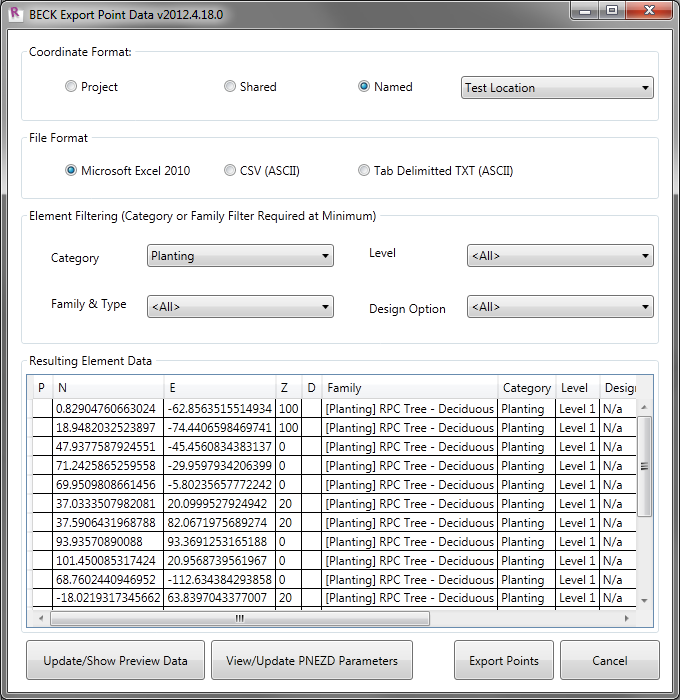


Figure 4 – BECK Export Point Data

## Tag Elements in Selected Views

This tool allows the user to filter elements by category and select a tag to place for each element of each selected family & type in as many views as they select.

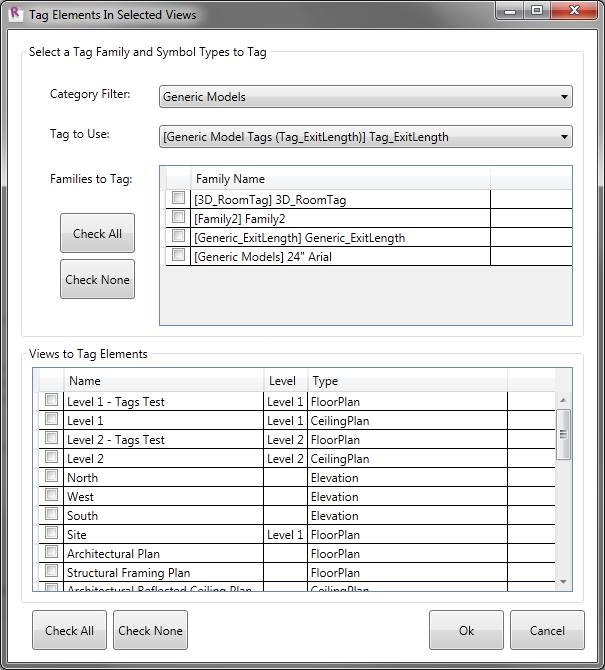


Figure - Tag Elements in Selected Views

## Data Manager

Importing data can be done from any of the same three file formats supported for reporting. The ElementID column is a required data member in order to match the element being edited with the data row in the external file. Once the ElementID is found within the model, the corresponding data is applied to the element. The user will be presented with a failure dialog displaying the ElementID’s that the system was unable to apply data changes to.

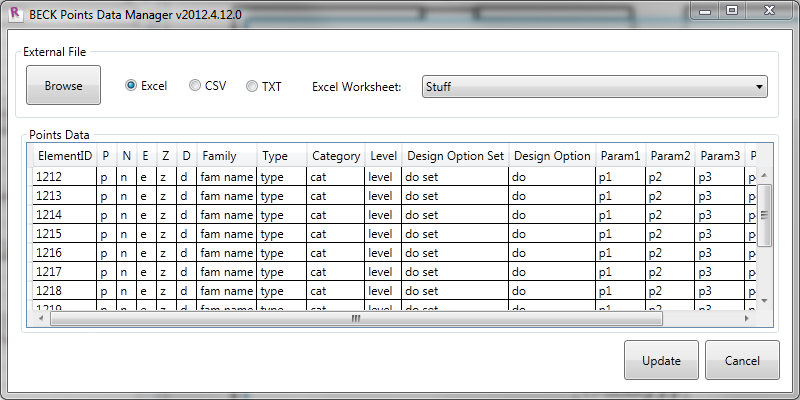


Figure 6 – BECK Points Data Manager (Excel Data Shown)

## Points from File

Importing PNEZD data from an external file supports Excel, CSV, or Tab Delimited TXT file formats. The N, E, and Z columns inputs are required in order to properly assemble a location object required to place the point. The “D” parameter is the only optional parameter value for importing points from external file.

The first row of the external file used to import the point families must be configured as the header with each column listed in order of PNEZD. The parameter name intended to hold the data for each parameter needs to be enclosed in brackets immediately after each column id as shown below (no spaces).

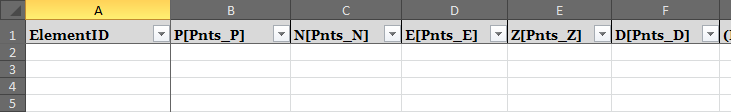


Figure 7- Sample Points Excel File

The image above shows a parameter name of “Pnts\_P” for the “P” parameter. This can be changed to whatever the user likes, but the parameter must first be loaded into the model or exist in the family selected to use as the point family prior to placing the points.

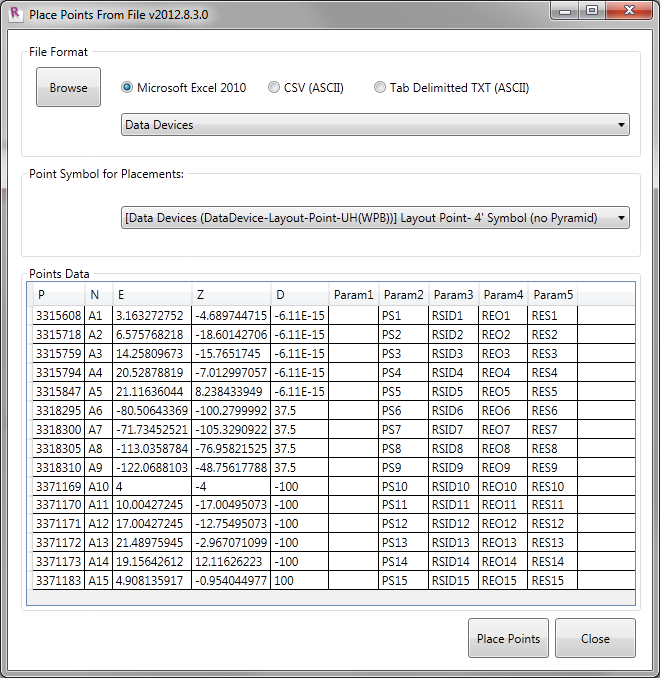


Figure 8 - Point Placement from File

When Excel is selected as an import format, the user will have to select the worksheet name to target the point data that they intend to import. Each of the named location systems found in the model are available as an import option as well.

## Parameter Manager

The parameter manager is used to assign default parameters based on category. These parameter assignments are used during export to fresh reports.

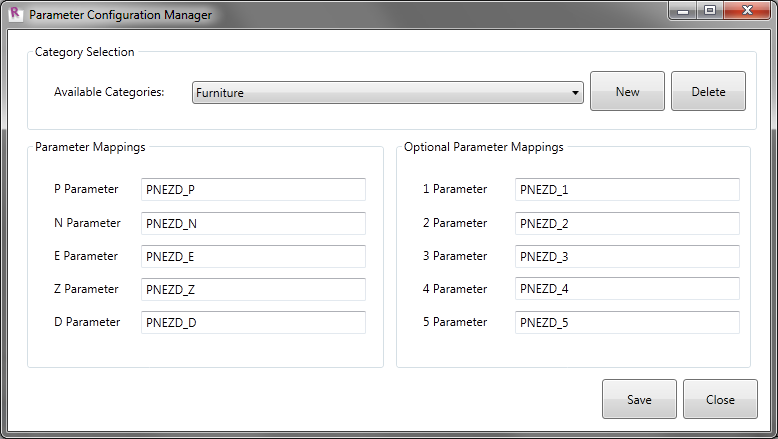


Figure 9 - Parameter Manager

## Manual Points

This command assumes access to each of the remaining miscellaneous functions. Each requires the user to first select the family that they wish to place along to the top of the form.

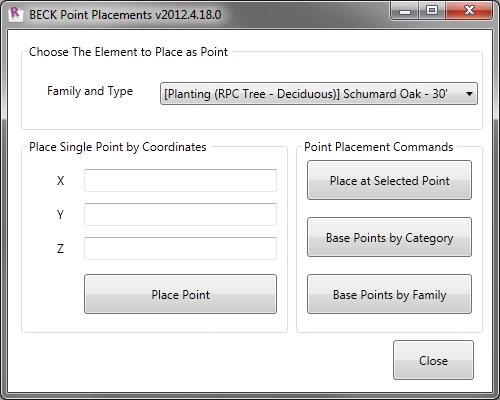


Figure - Manual Points

### Place Single Points by Coordinates

A point family is placed at the entered XYZ or selected location.

### Base at Selected Point

Place a point family at a point location selected on screen.

### Base Points by Category

Placing a point family at the base point of all elements of a chosen category will not check for pre-existing point families prior to placement, so it is important that the user first clean up any existing point families prior to running this option.

### Base Points by Family

Placing a point family at the base point of all elements of a chosen family type will not check for pre-existing point families prior to placement, so it is important that the user first clean up any existing point families prior to running this option.