AUTODESK UNIVERSITY

[AS502800]

Data-Driven Occupancy Load Workflow Using Revit

Clara Fonte Boa
EvolveLAB

Jim Greve
EvolveLAB

Learning Objectives

- Learn how to implement the Code Tools within their design and documentation processes.
- Challenge yourself to implement similar solutions.
- Learn how to implement a few new Revit tools in your own workflow.
  Learn about automating your Code documentation.

Description

Come to this session to discover how the occupancy load workflow is a live and data-driven tool that takes advantage of Revit software's recent advancements in order to facilitate the architect's code analysis and documentation journey. The workflow utilizes schedules, tags, shared parameters, and formulas to process raw data, calculate, and deliver the occupancy load directly to the code plans-while at the same time keeping the data flow between schedule and tags live and consolidated into a single source of truth.

Speaker(s)

Clara Fonte Boa is a design technologist with over five years of professional experience, a background in architecture, and a drive to use the available technology tools to the fullest. In the past three years, she worked at EvolveLAB, helping people to learn and implement BIM, Dynamo, and other computational technologies into their workflow, enabling a more efficient process that leaves more room for the creativity and the fun part of design to flourish. Clara also has a long history of fascination for topology, origami, complex structures, and geometry which inevitably brought her to the computational design world.
LinkedIn

Jim Greve is a BIM Manager on the EvolveLAB team with over twenty years of design and production experience in the AEC industry. Throughout that time Jim's roles have ranged from CAD Technician to BIM Manager, Architectural Designer to Terrestrial LiDAR Specialist, all while he gained an understanding of multiple disciplines and project types managing the technology and production of numerous design projects. Jim is driven to teach, encourage, and help maximize the potential within each client or team member he works with.
LinkedIn

About EvolveLAB

EvolveLAB is a BIM management and AEC App Development company specializing in interoperability, automation, and generative design tools for architects, engineers, contractors, and manufacturers. EvolveLAB's Mission is to leave the AEC industry better than when we came into it through disruption of inefficient processes and creating more streamlined workflows.

Find out more about us at www.evolvelab.io









Summary

In this class you will learn about the EvolveLAB Code Tools and look behind the curtains to see how it works.

We will go over each major Revit feature used, give a basic definition of how they work within a project, and mostly show **how** we combined complex calculations with Out Of The Box (OOTB) Revit tools to streamline and simplify the code review and documentation process.

It is our hope that after this class you will not only know how to create and review your Code Plans, but also feel empowered and inspired to implement these strategies to solve similar problems, optimize your process and further help your team.

What are the EvolveLAB Code Tools?

The EvolveLAB Code Tools is a set of Revit workflows and content that assists designers and architects in the creation and review of their Code Plans.

The Code Tools includes a series of custom schedules, tag families, color legends, view filters, custom parameters and other native Revit features creating a fully integrated and seamless process (no plugin or Dynamo script needed) to:

- Assign **Occupancy** and **Function of Space.**
- Calculate and document **Occupancy Load**.
- Calculate and document the **Number of Occupants**.
- Customize the **Path of Travel** documentation.
- Customize the **Assembly Fire Rating** documentation.
- And more.

The Origins

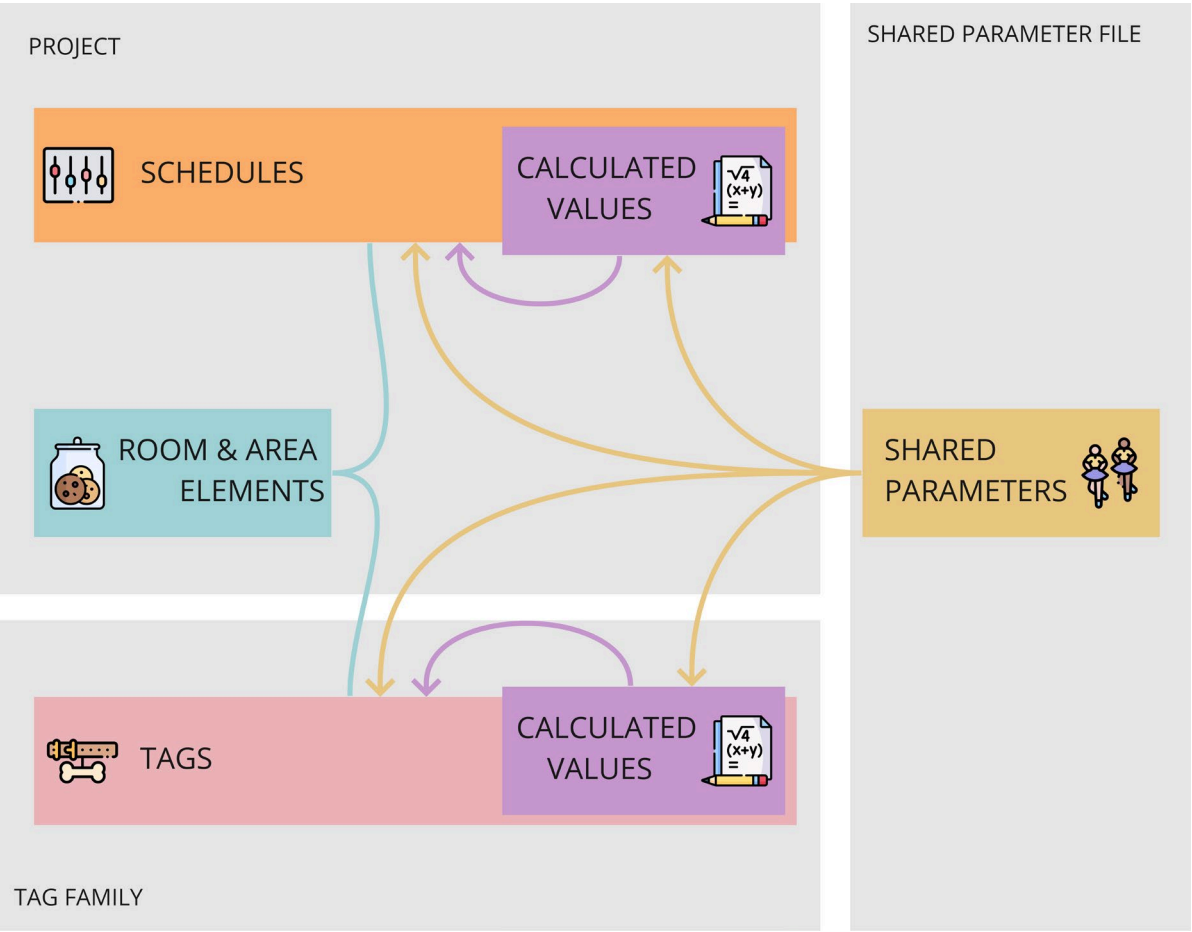
At EvolveLAB we work with a lot of clients, many of whom have expressed the need for a better standard for creating code drawings within Revit. This need led to the design of the Code Tools, which was created to assist in the calculation and documentation of building code analysis. It supports 2012, 2015, and 2018 International Building Code.

How the Code Tools uses Revit?

To create the process that ultimately calculates and documents the Occupancy Load and the Number of Occupants there are 5 major Revit Features used:

- Rooms & Areas Elements
- Schedules
- Tags
- Shared Parameters
- Calculated Values

The combination of these features allows us to gather data from the project, add some minimal inputs, track and review the calculations happening in the background and finally insert the results into the Code Plan sheets.



How it works:

The tool uses Rooms and Area elements to collect and store the data and reports it through schedules and tags. To do so, it uses Shared Parameters to synchronize the data between the schedule in the project and the tags. Finally, it uses calculated values parameters to host the complex formulas that calculates the results shown in both the schedules and the tags.

Rooms and Area Elements

What are Rooms and Area Elements?

A **Room** is a subdivision of space within a building model, based on room bounding elements such as walls, floors, roofs, and ceilings. An **Area** is a subdivision of space within a building model, typically on a larger scale than individual rooms. Areas are usually made up of several rooms.

Rooms and Areas are like containers that host data related to a specific portion of space in the project, data such as name, number, and square footage.

Rooms vs Areas in the Code Tools

When using the Code Tools, you can pick Room or Areas as the containers, and each has its advantages and disadvantages. So, it is up to the designer to decide which is more appropriate for each scenario and which option they will choose when using the Code Tools.

Rooms are typically easier and faster to work with. They:

- Can be quickly placed and annotate directly into Floor Plans.
- Automatically detect room bounding objects such as walls.

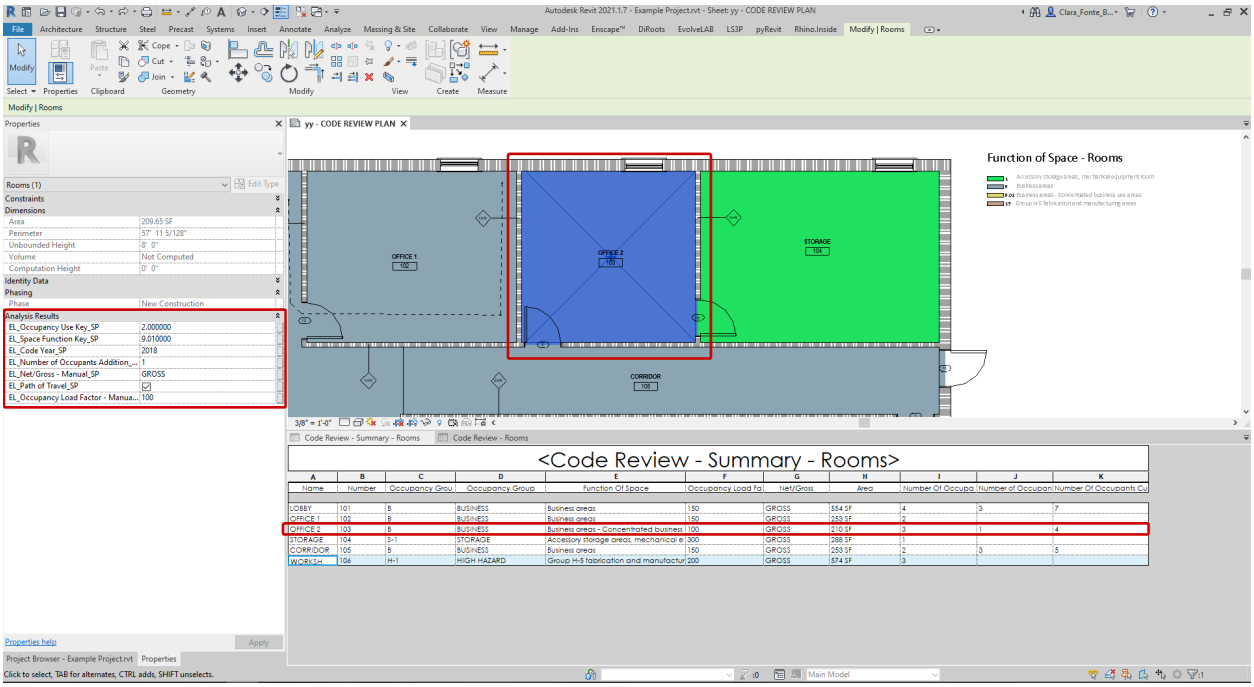
Areas are more versatile. They:

- Require manual drawing of all the boundaries.
- Are not constrained by any building elements like walls.
- Multiple different types of area element types can be placed in the same location, generating multiple Area Plans, each using different boundary lines to define its square footage.

How to use Rooms and Areas in the Code Tools?

As mentioned, the Room and Area elements are like containers for important parameters we will use in the Code Tools. Once instances are placed in the model, they automatically start working for us collecting model data such as square footage. By handling the Area and Room model instances we can also visualize and interact with all their parameters, inserting important inputs (Shared Parameters) and revising results (Calculated Values) that are calculated by the Code Tools for us.

These parameters can be seen in the Properties Panel when selecting the Room/Area Elements directly from the model space or within Schedules.



ROOM ELEMENT SELECTED IN PLAN. PARAMETERS DISPLAYED AT THE PROPERTIES PANEL AND SCHEDULE.

Schedules

What are Revit Schedules?

Revit schedules are usually created from the need to quantify and assign different values to the various elements in your Revit project. Revit schedules are a lot like a spreadsheet. They consist of columns and rows and the cells contain various data regarding the Revit elements. Custom parameters (aka fields) and formulas (Calculated Values) can be added to add extra information and to calculate new values from existing ones.

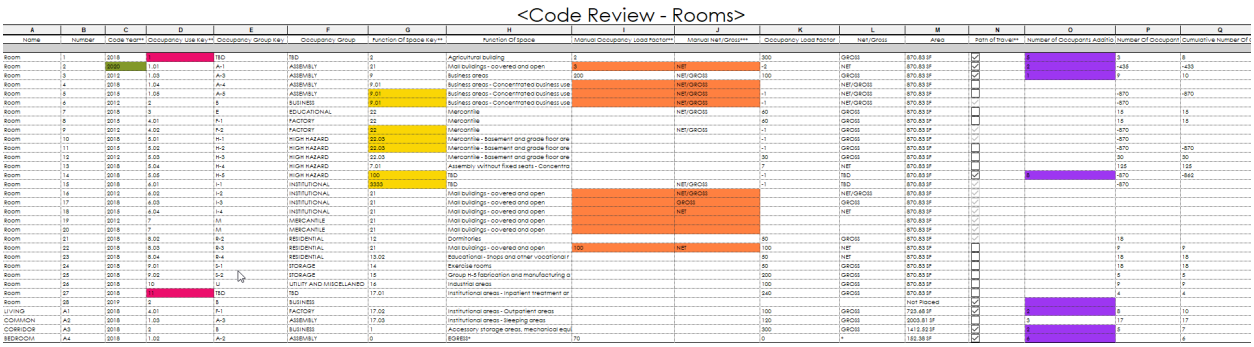
For our purpose today, we are using building component schedules, which list all model instances placed in the project, one in each row.

How to use the Code Tools Schedules?

When using the Code Tools, the schedule will become the main user interface for the designer/architect to enter information and interact with the tools. All the data will be contained in the rooms or areas instances, and easily accessed in the schedules.

To help the interaction, there are 2 different schedules: the **QA/QC Schedule,** and the **Documentation Schedule**.

The **QA/QC Schedule** is where you go to input the data, review the calculation, and confirm the results. This schedule is color coded with conditional formatting, helping you identify when and where manual inputs are necessary. And, as you insert inputs the Calculated Values parameters automatically calculate the results.



QA/QC SCHEDULE

On the other hand, the **Documentation Schedule** is not color coded and does not show all the information the **QA/QC Schedule** does. All the working fields are hidden, showing only the results needed for the Code Plans.

Code Review - Summary - Rooms

Name	Number	Occupancy Group Key	Occupancy Group	Function Of Space	Occupancy Load Factor	Net/Gross	Area	Number Of Occupants	Number of Occupants Addition	Number Of Occupants Cumulative
LOBBY	101	B	BUSINESS	Business areas	150	GROSS	554 SF	4	3	7
OFFICE 1	102	B	BUSINESS	Business areas	150	GROSS	253 SF	2		
OFFICE 2	103	B	BUSINESS	Business areas - Concentrated business use areas	100	GROSS	210 SF	3	1	4
STORAGE	104	S-1	STORAGE	Accessory storage areas, mechanical equipment room	300	GROSS	288 SF	1		
CORRIDOR	105	B	BUSINESS	Business areas	150	GROSS	253 SF	2	3	5
WORKSHOP	106	H-1	HIGH HAZARD	Group H-5 fabrication and manufacturing areas	200	GROSS	574 SF	3		

DOCUMENTATION SCHEDULE

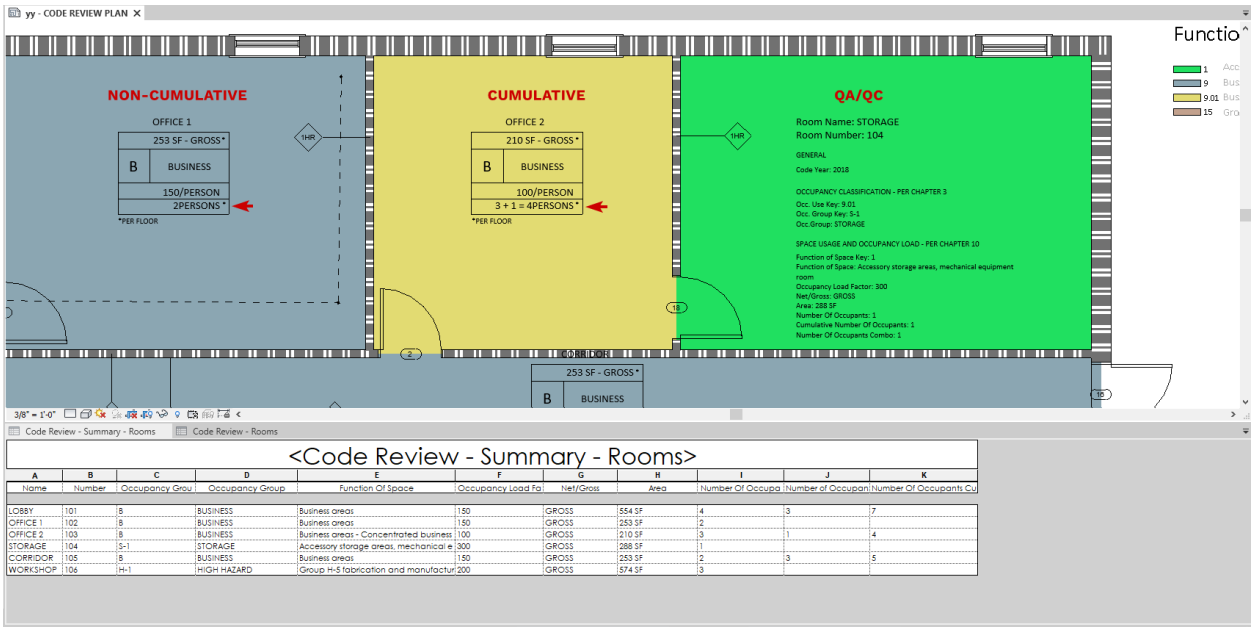
Tags

What are Tag families?

Revit Tag families are a type of annotation elements capable of pulling data from model objects to display them in drawings. Differently from plain text (that can be made to look like tags) Tag families' data (aka labels) are connected to the model object parameters. Thus, when the parameter values update, the Tag Labels will update as well, ensuring consistency through the model and drawings.

How to use the Code Tools Tags?

As part of the Code tools, we created a few custom Tags for Room and Area elements. Once you use the QA/QC Schedule, you can then place Tags in the Code Plans to display the results.



CODE TOOLS TAGS PLACED IN PLAN AND DOCUMENTATION SCHEDULE

Shared Parameters

What are Shared Parameters?

Shared Parameters are custom parameters that exists on a .txt file outside Revit and **help synchronize the data between project and family files**.

Usually, custom project parameters can only exist in a project and synchronize data specific to the project-based elements, while custom family parameters, similarly, only exists inside the family. Shared parameters on the other hand, are created outside and can be placed in both environments. Thus, allowing data to flow between project and families, mimicking the behavior of OOTB parameters.

Shared Parameters and the Code Tools

Shared Parameters is the Revit feature that allow us to connect the Room/Area elements, which are project-based to the custom Tag families. Synchronizing the Room/Area elements custom parameters to the labels in the Tag.

Calculated Values

What are Calculated Values?

Calculated Values are special type of custom parameters that **process data from other parameters through formulas**. It takes input parameters, processes the data, and return a new value as result.

Calculated Values in the Code Tools

The Code Tools contain multiple Calculated Values, which is in fact the machine that calculates and returns the results needed for the Code Plans.

To showcase how Calculated Formulas are used in the Code Tools, here are two examples.

Occupancy Group

In this example we have Occupancy Group formula, which is a combination of multiple conditional statements, testing whether the value inputted at the Occupancy Use Key parameter equals the values defined by the Occupancy Classification table. If the values match, the formula returns the equivalent Occupancy Group Key. For example, if the inputted key = 1.01, result = ASSEMBLY.

If statements take 3 inputs. A condition, a value to return if the condition is true and a second value to return if the condition is false. But, if you have many conditions like we have here, you can replace one of the return values with another if statement. In this example we keep adding a new condition to the false value until we reach the end of our list of keys. At the end, if none of the possible keys is inserted as the Occupancy Use Key, TBD will be returned.

EL_Occupancy Group_CV

```
if(EL_Occupancy Use Key_SP = 1.01, "ASSEMBLY", if(EL_Occupancy Use Key_SP
= 1.02, "ASSEMBLY", if(EL_Occupancy Use Key_SP = 1.03, "ASSEMBLY",
if(EL_Occupancy Use Key_SP = 1.04, "ASSEMBLY", if(EL_Occupancy Use Key_SP
= 1.05, "ASSEMBLY", if(EL_Occupancy Use Key_SP = 2.00, "BUSINESS",
if(EL_Occupancy Use Key_SP = 3.00, "EDUCATIONAL", if(EL_Occupancy Use
Key_SP = 4.01, "FACTORY", if(EL_Occupancy Use Key_SP = 4.02, "FACTORY",
if(EL_Occupancy Use Key_SP = 5.01, "HIGH HAZARD", if(EL_Occupancy Use
Key_SP = 5.02, "HIGH HAZARD", if(EL_Occupancy Use Key_SP = 5.03, "HIGH
HAZARD", if(EL_Occupancy Use Key_SP = 5.04, "HIGH HAZARD",
if(EL_Occupancy Use Key_SP = 5.05, "HIGH HAZARD", if(EL_Occupancy Use
Key_SP = 6.01, "INSTITUTIONAL", if(EL_Occupancy Use Key_SP = 6.02,
"INSTITUTIONAL", if(EL_Occupancy Use Key_SP = 6.03, "INSTITUTIONAL",
if(EL_Occupancy Use Key_SP = 6.04, "INSTITUTIONAL", if(EL_Occupancy Use
Key_SP = 7.00, "MERCANTILE", if(EL_Occupancy Use Key_SP = 8.01,
"RESIDENTIAL", if(EL_Occupancy Use Key_SP = 8.02, "RESIDENTIAL",
if(EL_Occupancy Use Key_SP = 8.03, "RESIDENTIAL", if(EL_Occupancy Use
Key_SP = 8.04, "RESIDENTIAL", if(EL_Occupancy Use Key_SP = 9.01, "STORAGE",
if(EL_Occupancy Use Key_SP = 9.02, "STORAGE", if(EL_Occupancy Use Key_SP =
10.00, "UTILITY AND MISCELLANEOUS", "TBD"))))))))))))))))))))))))))
```

OCCUPANCY CLASSIFICATION - PER CHAPTER 3					
OCCUPANCY USE KEY	OCCUPANCY GROUP KEY	OCCUPANCY GROUP	APPLICABLE CODE		
			2012	2015	2018
1.01	ASSEMBLY	A-1	Y	Y	Y
1.02	ASSEMBLY	A-2	Y	Y	Y
1.03	ASSEMBLY	A-3	Y	Y	Y
1.04	ASSEMBLY	A-4	Y	Y	Y
1.05	ASSEMBLY	A-5	Y	Y	Y
2.00	BUSINESS	B	Y	Y	Y
3.00	EDUCATIONAL	E	Y	Y	Y
4.01	FACTORY	F-1	Y	Y	Y
4.02	FACTORY	F-2	Y	Y	Y
5.01	HIGH HAZARD	H-1	Y	Y	Y
5.02	HIGH HAZARD	H-2	Y	Y	Y
5.03	HIGH HAZARD	H-3	Y	Y	Y
5.04	HIGH HAZARD	H-4	Y	Y	Y
5.05	HIGH HAZARD	H-5	Y	Y	Y
6.01	INSTITUTIONAL	I-1	Y	Y	Y
6.02	INSTITUTIONAL	I-2	Y	Y	Y
6.03	INSTITUTIONAL	I-3	Y	Y	Y
6.04	INSTITUTIONAL	I-4	Y	Y	Y
7.00	MERCANTILE	M	Y	Y	Y
8.01	RESIDENTIAL	R-1	Y	Y	Y
8.02	RESIDENTIAL	R-2	Y	Y	Y
8.03	RESIDENTIAL	R-3	Y	Y	Y
8.04	RESIDENTIAL	R-4	Y	Y	Y
9.01	STORAGE	S-1	Y	Y	Y
9.02	STORAGE	S-2	Y	Y	Y
10.00	UTILITY AND MISCELLANEOUS	U	Y	Y	Y

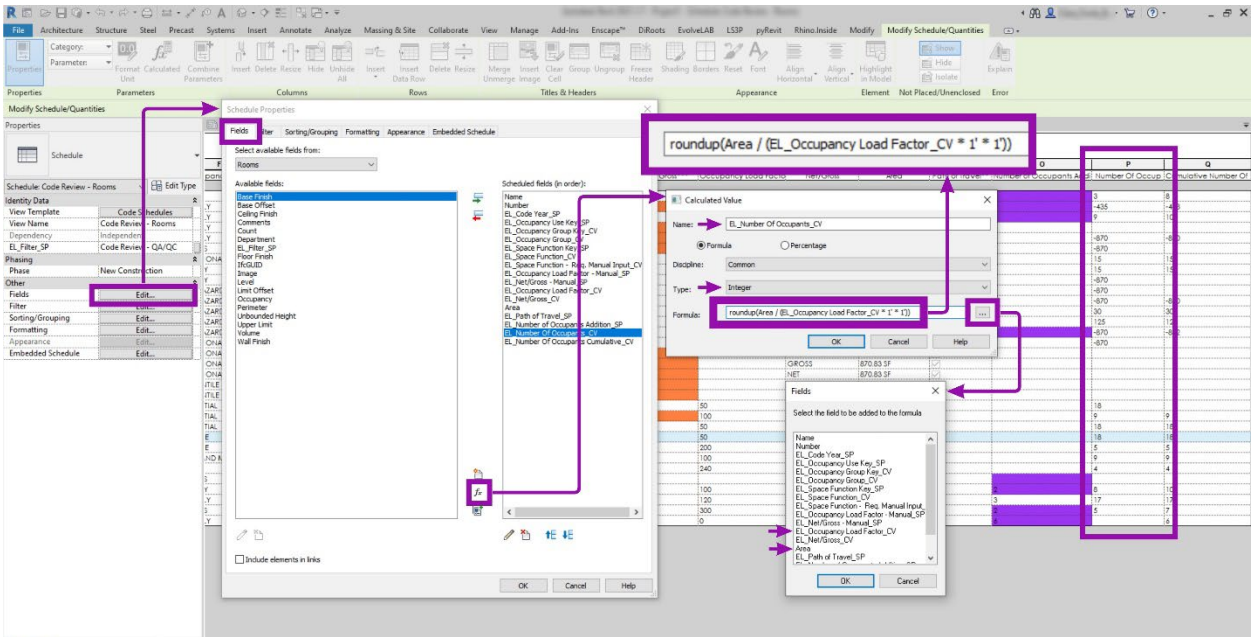
OCCUPANCY GROUP CALCULATED VALUE FORMULA AND REFERENCING TABLE

Number of Occupants

In this example we show how the Number of Occupants formula is created in Revit.

- From the Schedule view, we go to edit the fields to open the Fields tab in the Schedule Properties window.
- From there we click in the calculated value button to create a new Calculated Value parameter.
- In thin new pop-up we can then create the formula.

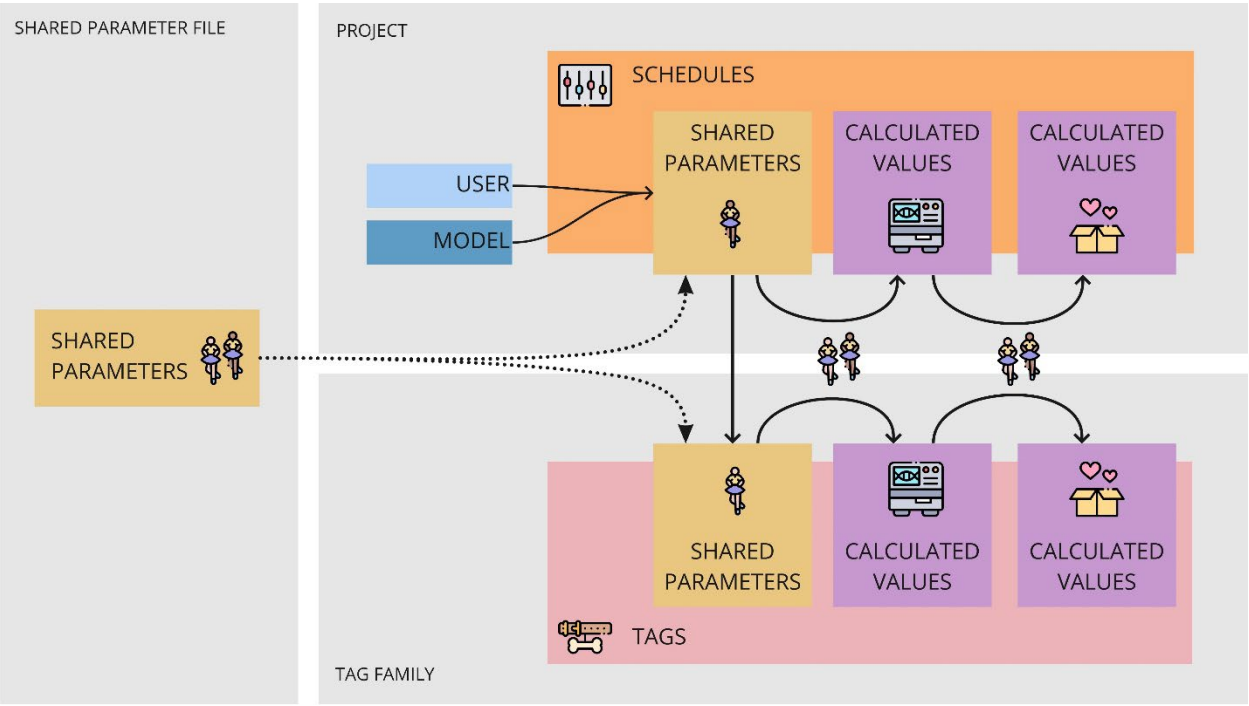
The formula in this example is an arithmetic equation that divides the Area (unit: square foot) by the Occupancy Load Factor (unit: integer). The formula also rounds the result up and divides it by 1 foot twice to ensure the result is always an integer (which is the unit defined for this parameter.



How do all the features work together?

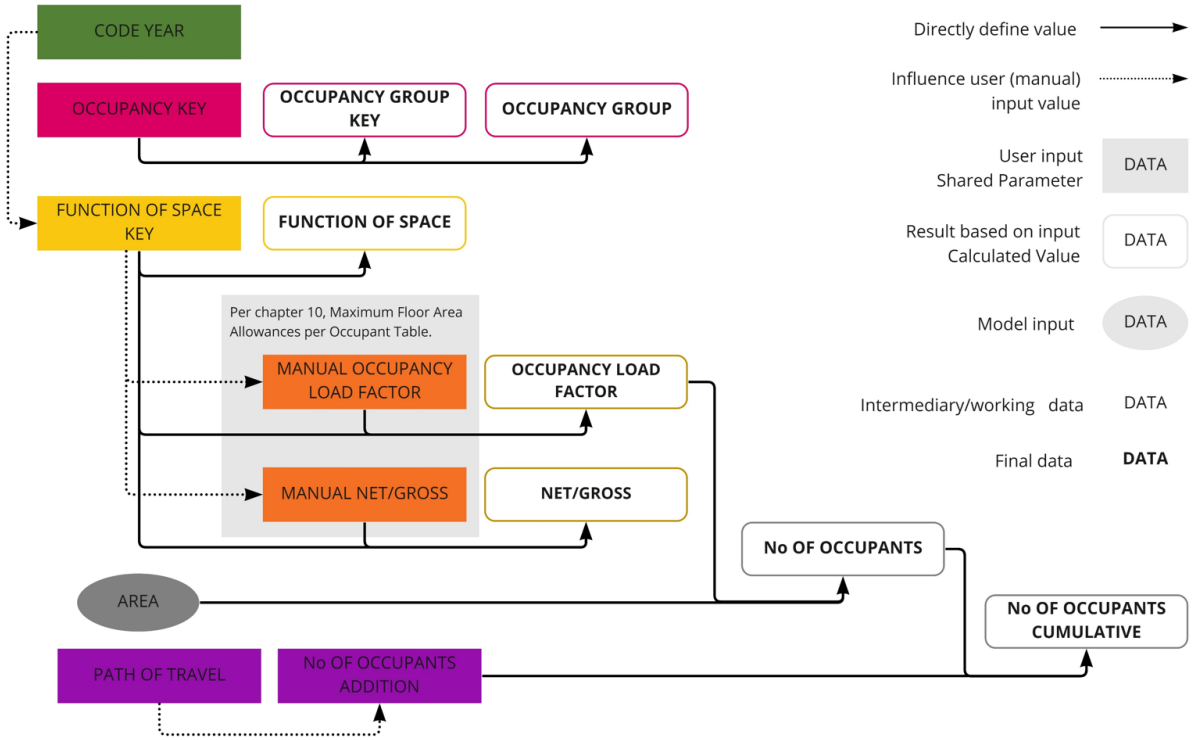
As we learned so far, the Code Tools:

- Takes user and model inputs through OOTB and Shared Parameters.
- Provides the QA/QC Schedule as an interface to input values and check the results.
- Uses Shared Parameters to ensure the Schedules, and the Tags data are synchronized through the whole process.
- Uses Calculated Values placed in both the schedules and the tags to do the calculations that give us the results we need.
- Provides the Documentation Schedule and the Tags to display the results in the Code Plan sheets.



What is the Code Tools Calculation Process?

Now that we understand the mechanisms used by the Code Tools, lets dive in a little further and see how the process takes shape and flow from parameter to parameter.



In summary

INPUTS

- User inputs values for:
  - **Code Year**
    (2012, 2015, 2018)
  - **Occupancy Use Key**
    (Defined by the Occupancy Classification table)
  - **Function of Space Key**
    (Defined by the Space and Occupancy Load table)
  - **Path of Travel**
    (Whether the room/area element should consider the Number of Occupants Addition or not)
  - **Number of Occupants Addition**
    (Defined by the number of Occupants flowing from other rooms/areas)
  - **Manual Occupancy Load Factor** and **Manual Net/Gross**
    (Manual inputs whenever values are not defined by the Maximum Floor Area Allowances Per Occupant table from IBC's chapter 10)
  -
- The model automatically inputs
  - **Area**
    Square footage of each room and area elements placed in the model.

NOTE: The Occupancy Key and the Function of Space Key are numerical values defined by a table as work around because Revit can't use text as formula inputs.

PARAMETER RELATIONSHIPS

- The **Code Year** influences applicable
  - **Occupancy Use Key**
  - **Function of Space Key**
- The **Occupancy Use Key** drives the results of:
  - **Occupancy Group Key**
  - **Occupancy Group**
- The **Function of Space Key**

  Defines the results of:
  - **Function of Space**
  - **Occupancy Load Factor**
  - **Net/Gross**

  Influences the need for:
  - **Manual Occupancy Load Factor**
  - **Manual Net/Gross**
  - value, and influences whether the **Occupancy Load Factor** and **Net/Gross** results require a manual input.
- **Occupancy Load Factor** and **Area** defines:
  - **Number of Occupants**
- **Number of Occupants** and **Number of Occupants Addition** defines:
  - **Number of Occupants Cumulative**

Extra Tools and Features

Color Legend

To help illustrate the code analysis results the Code Tools also provides a custom Legend to annotate the Rooms and Areas Function of Space.

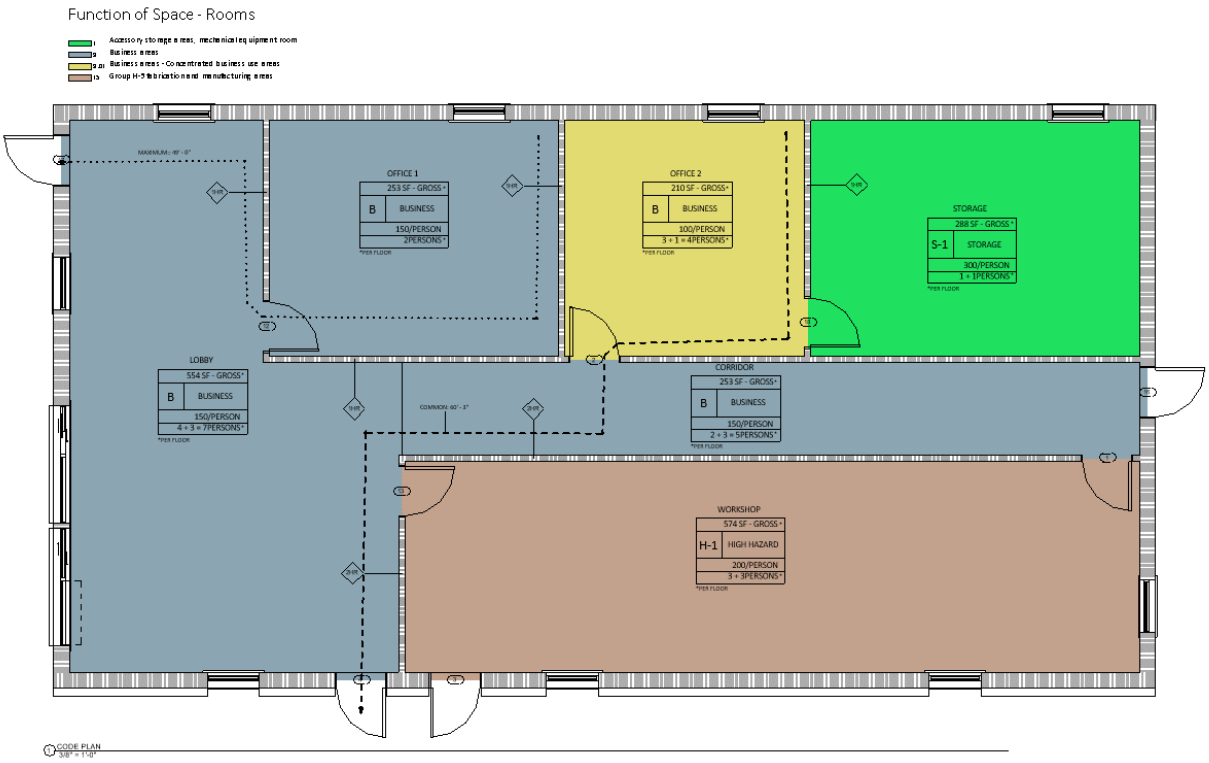
The Legend is based on a custom a Color Scheme created in the project, overlayed by a customized schedule.

Path Of Travel

The Code Tools also relies on the OOTB Path of Travel Revit Tools and provides custom Line Styles and Tags to help annotate different types of paths.

Fire Rating Filter Overrides

The Code Tools contains customized View Templates and View Filters designed to annotate the Walls Fire Rating. These filters are pre-set in the project and will display a custom fill pattern corresponding to the wall's Fire Rating value.



CODE PLANS EXAMPLE

Usefull Links

- EvolveLAB Code Tools
  - https://www.evolvelab.io/product-page/revit-code-tools
- Calculated Values and Formulas in Revit
  - https://knowledge.autodesk.com/support/revit/learn-explore/caas/CloudHelp/cloudhelp/2020/ENU/Revit-Model/files/GUID-BA812528-8000-44A5-B53F-137B3F642107-htm.html
- Shared Parameters and Tag
  - https://knowledge.autodesk.com/support/revit/learn-explore/caas/CloudHelp/cloudhelp/2018/ENU/Revit-Model/files/GUID-2A2FD10E-0F50-41B8-80DF-972EF895A275-htm.html

Copyright Mentions

Icons

- Freepik – Flaticon
  - Cookie icons created by Freepik - Flaticon
  - Adjust icons created by Freepik - Flaticon
  - Slider icons created by Freepik - Flaticon
  - Math icons created by Freepik - Flaticon
  - Lab icons created by Freepik - Flaticon
  - Love icons created by Freepik - Flaticon
  - Choreography icons created by Freepik - Flaticon