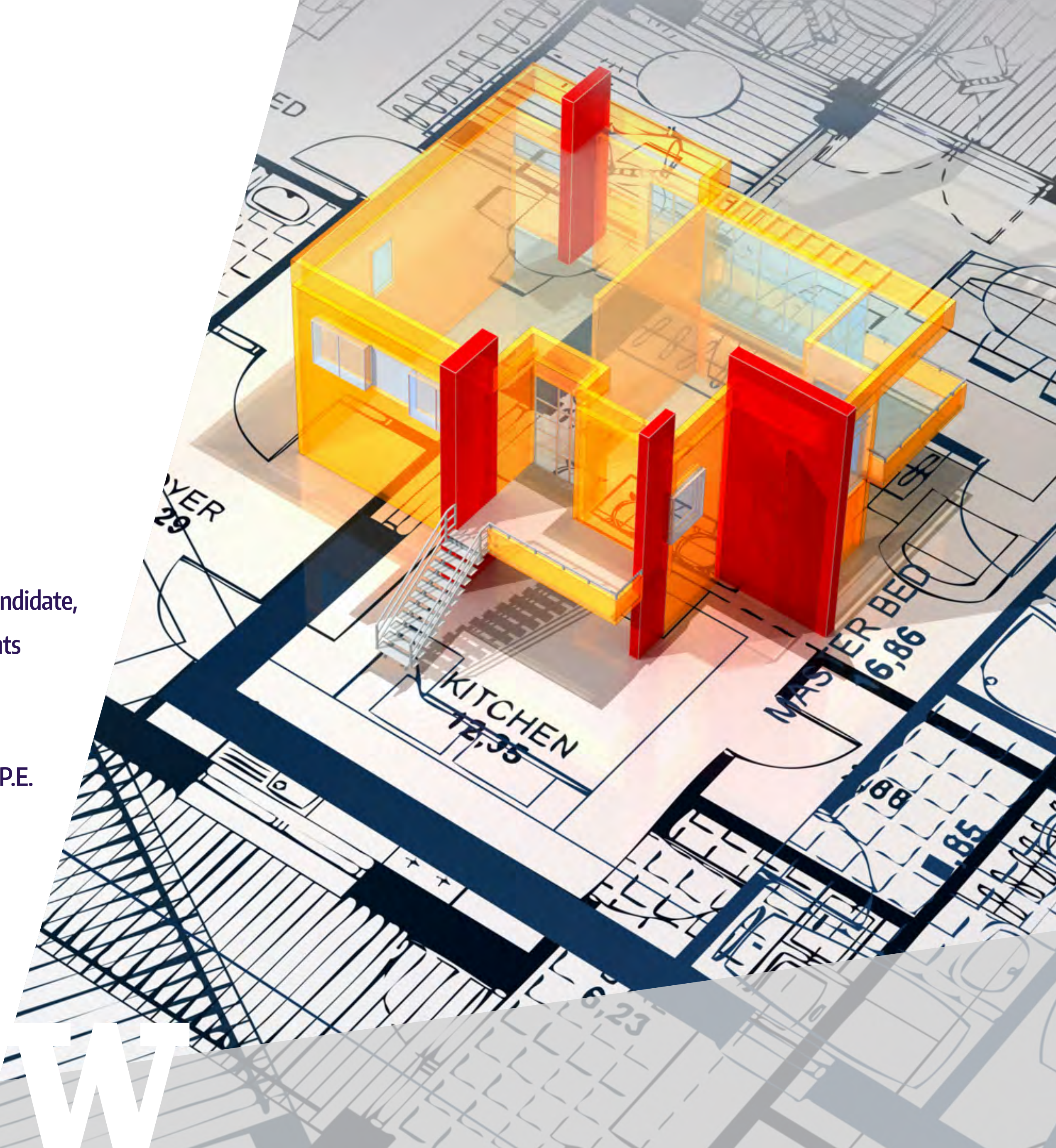
CENTER FOR EDUCATION +
RESEARCH IN CONSTRUCTION
DEPARTMENT OF CONSTRUCTION MANAGEMENT

# REBASELINING ASSET DATA FOR EXISTING FACILITIES AND INFRASTRUCTURE




By Hamid Abdirad, Ph.D. Candidate,
College of Built Environments

Supervising Faculty:
Carrie Sturts Dossick, Ph.D. P.E.

# ABOUT CERC

The Center for Education and Research in Construction (CERC) is a locus of research, scholarship and discovery in the UW's Department of Construction Management and allied disciplines of architecture, engineering and real estate. Focused on the people and practices of a dynamic, innovative construction industry, CERC develops new concepts and innovative solutions as well as improved methodologies for design, construction and operations. With active labs focused on Safety and Health, Project Delivery and Management, Virtual Design and Construction, Infrastructure Development, and Sustainable Built Environments the CERC faculty are not only experts and researchers in a wide array of topics but also lead the field in translating that expertise into excellent construction education practices and pedagogy to train tomorrow's construction professionals.

**http://cm.be.uw.edu/cerc/**

# TABLE OF CONTENTS

# EXECUTIVE SUMMARY

The redundant and costly effort for data collection, re-entry, and revalidation in asset management practices imposes billions of dollars per year on the U.S. capital facilities industry. Although recent technology developments and information delivery schema (e.g., COBie) have tackled the need for systematic delivery of asset data for new facilities, collecting reliable and verifiable data for existing facilities and infrastructure has been a significant challenge. This is because (1) data requirements for asset management change and become more demanding, (2) existing assets go through modifications and renovations often without appropriate documentation, and (3) available data on existing assets may be inaccurate or inconsistent.

In this report, we introduce rebaselining as a workflow for collecting and verifying data for existing assets based on evolving or emerging data needs or changes in the status of assets. We conducted two action research projects in two public owner organizations in the U.S. to explore the rebaselining tasks that owners need to define and the decisions they need to make to mitigate the challenges of collecting data for existing assets. This study frames rebaselining in four phases: 1) preparing technology enablers, 2) collecting asset data from existing documents, 3) field verification, and 4) updating asset management databases. The results of this work set the foundation for implementing rebaselining workflows and technologies by (1) mapping rebaselining business processes in each phase, (2) listing technological affordances required in these processes, and (3) showing how practitioners can customize and leverage conventional information modeling platforms to implement rebaselining workflows.

# 1 INTRODUCTION

Asset management is the practice of managing and allocating resources to align the life-cycle status of assets to the financial and functional goals of organizations.[1] Asset management determines what assets an organization owns; what conditions they are in, how they are expected to perform over time; and how and when they should be maintained, repaired, or replaced.[2] The significant challenges to asset management emerge when available asset data are inadequate, inaccurate, or unable to support asset management processes. Prior research shows that this is an ongoing challenge in the industry because inadequate information and data interoperability issues cost capital facilities billions of dollars per year in the U.S. due to inefficient business processes and redundant efforts for asset data collection, re-entry, and validation.[3] Although recent relevant developments such as Construction-Operation Building information exchange (COBie) specifications have addressed the need for a systematic approach to asset data collection and hand-over through the design and construction of new buildings, there has been limited research and industry practice documenting how to aggregate reliable asset data for existing assets. This is especially important because:

1) Data requirements for asset management evolve due to advances in asset management practice and standards, which have become more demanding in the past several years.[4]

2) Facilities and infrastructure undergo modifications (e.g., adding, removing, replacing assets) often without proper documentation. Therefore, existing data, drawings, and documents may become obsolete.

1. Too, E. G. (2010). A Framework for Strategic Infrastructure Asset Management. In J. E. Amadi-Echendu, K. Brown, R. Willett & J. Mathew (Eds.), *Definitions, Concepts and Scope of Engineering Asset Management* (Vol. 1, pp. 31-62). London: Springer London.

2. Institution of Structural Engineers. (2011). *Design Recommendations for Multi-Storey and Underground Car Parks* (4th Edition). London: Institution of Structural Engineers.

3. Gallaher, M. P., O'Connor, A. C., Dettbarn, J. L., & Gilday, L. T. (2004). *Cost Analysis of Inadequate Interoperability in the U.S. Capital Facilities Industry.* Washington D.C.: National Institute of Standards and Technology; U.S. Department of Commerce.

4. Kapelan, Z., Banyard, J. K., Randall-Smith, M., & Savić, D. A. (2011). Asset Planning and Management. In D. A. Savić & J. K. Banyard (Eds.), *Water Distribution Systems*. London: ICE Publishing.

3) In terms of content and formatting, inconsistencies between the required asset data, data received for new assets, and available data for existing assets lead to disruptions in asset management activities.

4) Inaccurate data (e.g., quantities and attributes) and poor referencing systems (e.g., IDs, tags, serial numbers) are prevalent in existing asset datasets due to error-prone manual processes.

5) Using information models (e.g., Building Information Models) for aggregating and tracking asset data is financially feasible in new projects since these models are used for many different processes, such as design authoring and construction documentation. Existing facilities and infrastructure may not have digital models to facilitate data collection, and creating full information models only for asset management can be infeasible in terms of cost and time.

These issues highlight the need for a systematic approach to develop reliable asset data for operating and maintaining existing facilities and infrastructure. Through two action research studies in collaboration with asset management practitioners in the public domain, our research lab introduced rebaselining as a workflow for collecting and/or verifying asset data for existing assets. In summary, this report highlights the rebaselining tasks that owners need to define, challenges owners might face during the course of rebaselining, and decisions they need to make to mitigate the challenges of collecting and working with data for existing assets. Additionally, this report demonstrates two technical processes through which practitioners can customize and use Navisworks and Revit as a conventional information modeling platform for efficient rebaselining workflows.

## 1.1 Asset Classification Systems

Asset classification systems are standards or custom approaches used for assigning assets into different groups based on their characteristics (i.e., creating a taxonomy). For example, one may consider conventional building and infrastructure systems, called asset systems in this report, as a high-level characteristic that facilitates classifying assets (e.g., mechanical, electrical, structural, and Plumbing as asset systems). Each asset system may include many different asset categories that are distinguished based on their functional role in the asset systems (e.g., ducts, air handling units, and air diffusers are some asset categories in a mechanical system). Each asset category may include different asset types (e.g., Power Generator Type A or Power Generator Type B). Asset attributes establish differences between asset types through providing information about (1) properties or functional characteristics of assets (e.g., voltage and phases in a power generator), and (2) attributes assigned to the assets by industry standards or the actors involved in the supply chain (e.g., model number, manufacturer). Each asset system in a building or infrastructure may include one or more asset instances of a specific asset type. Each asset instance may have additional asset attributes that can distinguish it from other asset instances of the same type (e.g., mark or tag number, location, age, condition). These asset systems, categories, types, instances, and attributes all form asset data that are required for asset management interventions. Asset owners collect and maintain asset data in a comprehensive dataset called the *asset inventory*.

# 2 SUMMARY OF RESEARCH INTERVENTIONS AND RESULTS

## 2.1. Project 1: Public Transportation Agency

Sound Transit (ST) is a regional public transportation agency that procures, operates, and manages city and county-wide public transportation facilities and infrastructure (e.g., stations, parking structures, light-rail, railway, and maintenance facilities). The age of the facility investigated in this action research study did not exceed 20 years. This agency has been commissioning BIM models for new facilities since 2013, although acquiring asset management data was not planned to be a part of project procurement until 2016. For existing buildings, asset management staff internally collected and aggregated asset data in data spreadsheets and in a CMMS. In 2015, the asset planning and programing department realized that previous asset data specifications must change to streamline the many different processes within the department. These processes include, but were not limited to, conditions assessment of assets, design review processes, capital planning and programming, creating life-cycle cost plans, and financing. For this reason, data requirements for asset data were deemed to be more detailed than asset databases were able to provide at the time. For example, in addition to a fixed set of conventional asset data attributes for all assets (Table 1), ST needed quantity and cost data on all asset categories; including linear assets such as pipes, ducts, roads, and railways to calculate future replacement cost forecasts. These revisions in existing data specifications highlighted that the agency did not have such data readily available for its large portfolio of existing assets. The analysis of existing data on quantities of assets revealed that designers, contractors, and in-house estimators used different taxonomies and work breakdown structures that did not satisfy asset management requirements. One of the goals for this effort was to develop a rebaselining workflow that facilitated collecting, aggregating, and verifying asset data of existing buildings to support asset management activities in this agency.

| | |
|---|---|
| **Asset systems** | Structure (architectural and structural assets- e.g., roofs, walls, guardrail); Site and Civil (landscape and hardscape assets – e.g., roadways, fencing, planting area); Equipment and Fixtures (e.g., HVAC, Electrical, Communication, Lighting, Fire Detection, Fire Suppression, Plumbing, Signage). |
| **Required asset attributes for all asset categories** | Asset Category, Asset ID, Field Asset Tag, Asset Assembly, Asset Description, Facility/Area, Location in Facility Area, Manufacturer/Vendor, Model #, Serial #, Quantity, Unit of Measure, Area Served |

*Table 1: Summary of asset data requirements for Sound Transit.*

The research team selected a 350,000 square-foot bus station and parking facility (built in 2004) for rebaselining workflow development. This facility accommodated many different structural, civil, site, and equipment asset categories that were listed in the data specifications. There was no BIM/CAD file available for this facility (drawings and manuals were all in PDF format), and the existing asset data for this facility were partial and could not satisfy the latest data requirements. In the search for a technology platform that could support the rebaselining workflow in this agency, the team chose Autodesk Navisworks based on the following considerations:

- Navisworks had a quantification tool that could support taking-off quantities of assets with different units of measure (e.g., each, linear foot, square foot) without building full information models. This tool supported using 2D drawings in the quantification process as the users could draw mark-ups (e.g., points, lines, polygons) to highlight different asset instances and quantify them as an overly.

- Since all asset categories in this agency had the same set of data attributes, an object-based information modeling platform was not required in this process. All asset categories and types could be treated in the same way in terms of data attributes.

- Since this agency used Autodesk Navisworks for design and construction projects, this rebaselining workflow did not require any investment for new tools and services.

In order to prepare Navisworks as the data aggregator, the team converted the asset data specification to an XML format data file in order to facilitate importing asset categories into the Quantification tool in Navisworks. As custom 'asset attributes' (except for quantities) were not pre-loaded in the quantification tool, the author needed to directly edit the default XML Configuration file on Navisworks installation to add these attributes into the quantification tool. In the planned workflow, users imported the XML catalog and existing drawings (PDF files) into a new Navisworks file using the Quantification and Sheet Browser tools. The quantification tool consisted of a list of asset categories and a data worksheet (Figure 1:2). After scaling the drawing sheets to 1:1, users could select an asset category from the list and mark-up drawings to document relevant assets (Figure 1:1). With each mark-up, Navisworks created a new object or asset in the data worksheet. These mark-ups were scale-aware in this process and they retrieved quantities of assets shown as an overlay to the PDF document. The quantities were automatically reported into the data worksheet (Figure 1:3; users could override quantities if they wanted to), making this process accurate, reliable, and verifiable through a visual confirmation of the mark-up with the drawings. After visualizing assets with mark-ups on drawings and having a list of assets in the data worksheet, users could type other attributes into the worksheet to aggregate data for each and every asset (Figure 1). By selecting a mark-up on drawings, users could access attached data attributes from the data worksheet. By clicking on asset data in the data worksheet, the related markup would be highlighted and shown in the viewport. Users could choose different colors for mark-ups of each asset category to facilitate finding and tracking them visually during the process.

After creating the asset data model in Navisworks, the team used the file for field verification of asset locations and data attributes. Having underlay drawings and overlay mark-ups along with asset IDs streamlined the field verification process as all required data could be verified across different data sources in a single interface. In the field verification process, the team attempted to identify missing data, verify or update data extracted from documents, and document field modifications by adding or modifying mark-ups or adding comments to the data worksheet. These mark-ups could serve as up-to-date references for documenting the revised status of managed assets.

The field verification process revealed that tags or plates on physical assets might also need to be updated due to possible inconsistencies between data sources or missing identification data on assets. For example, team members might need to place asset ID and name tags on assets when they find that existing tags are illegible or misplaced. After the field verification process and updating the asset data model, asset inventory data were exported to the Computerized Maintenance Management System (CMMS) for record keeping and work-order processing. The Navisworks file itself was also used for accessing documentations, verifying asset data, and documenting the field status of assets when needed.

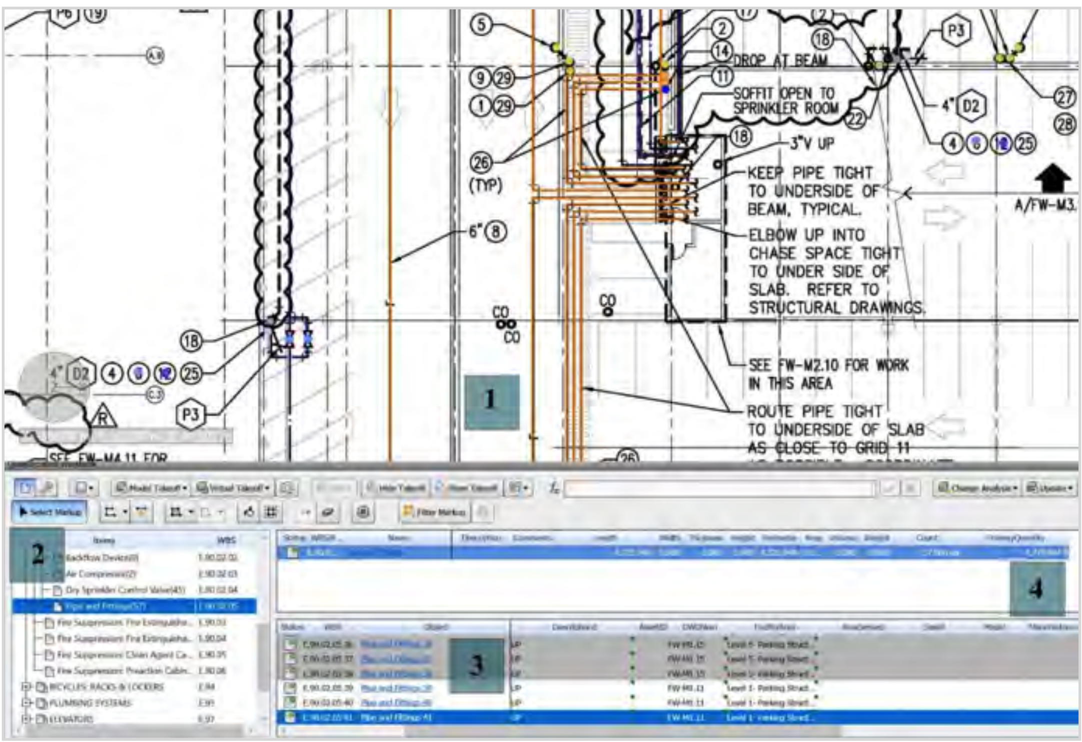



*Figure 1 – User interface in Navisworks for rebaselining. 1: Mark-ups on drawings; 2: Asset categories; 3: Asset instances and attributes in data worksheet; 4: Total quantity for the selected asset category.*

## 2.2. Project 2: Higher Education Institutional Campuses and Facilities

The University of Washington (UW) is a public institutional owner that procures, operates, and maintains different building types on multiple campuses for higher education and research. At the time of this action research, the portfolio of assets included more than five hundred buildings that exceeded 25 million square feet. The asset management activities that UW implemented included asset maintenance and asset repair, for which asset lists and attributes were collected and aggregated into a CMMS. As the UW owns a large portfolio of buildings with different ages (up to 120 years old), data collection regimes and requirements have been subject to many changes throughout the years. Before 2011, asset data were collected internally by facility management staff. Since adopting COBie in 2011, design and construction teams have collected and handed over asset data for some of the new buildings. COBie implementation was an impetus to identify and establish asset categories and attributes that are required for maintenance and repair. However, UW asset data specifications differ from the standard COBie specification. Each asset category not only requires common data attributes defined in COBie standards (e.g., manufacturer, space), but also requires some custom data attributes that are specific to each asset category (e.g., capacity needed for VAV boxes; see Table 2).

| | |
|---|---|
| **Assets systems (and asset categories as examples)** | Equipment of Building Systems (Mechanical, Electrical and Communication, Plumbing); Architectural and Structural Components (e.g., Ceiling Tile System, Floor System, Roof System, Doors, Fixed Seats); Landscape and Hardscape Elements (e.g., Artworks, Light Poles, Trash Compactor). |
| **Common attributes for all assets (modified COBie)** | Contacts info, Facility info, Floor Level, Space, Zone, Asset Name, Asset Category, Description, Asset Type, Manufacturer, Model Number, Warrant Description, Warranties Duration, Warranty Guarantor, Replacement Cost, Expected Life, Location, Connected Systems, Documents, Spare, GIS Coordinate, Resource, Job, etc. |
| **Examples of custom defined attributes for generators** | Year Built, License Plate #, Air Filter, Coolant Capacity, Engine Oil Capacity, Engine Serial #, Engine Type, Fan Belt, Fuel Filter, Grease, Hydraulic Oil Capacity, Hydraulic Oil Type, Oil Filter, Oil Type, Tire Size, Transmission Oil Capacity, Transmission Oil Filter, Type, Valve Specs, Power Supply, Capacity, Electric Generator Efficiency, GFCI Capable, Number of Sources, Maximum Power Output, Start Current Factor, Fuel Type, Fuel Storage, Fuel Capacity, Operating Rpm Limits, Engine Cooling Type, Engine Size. |
| **Common attributes for all assets (modified COBie)** | Contacts info, Facility info, Floor Level, Space, Zone, Asset Name, Asset Category, Description, Asset Type, Manufacturer, Model Number, Warrant Description, Warranties Duration, Warranty Guarantor, Replacement Cost, Expected Life, Location, Connected Systems, Documents, Spare, GIS Coordinate, Resource, Job, etc. |

*Table 2: Summary of asset data requirements for UW.*

For this action research, the team selected a 265,000 square-foot research facility (built in 2006) as a pilot case for designing, testing, and refining the workflow. This facility was deemed a reasonable choice in terms of complexity, because (1) BIM and COBie data were not commissioned during its design, construction, or handover, and (2) the building had many different asset categories that included specialty equipment in addition to conventional building systems. The team decided to use a conventional BIM platform that would support (1) the creation of asset geometries, (2) the attachment of asset data to geometries, (3) viewing drawings as underlay/overlay sheets, and (4) linking other project documents to assets (e.g., operation and maintenance manuals). The team selected Autodesk Revit for the workflow due to the following considerations:

- Since the owner used Autodesk Revit for accessing models of new and recently built projects, the team decided to use Revit for rebaselining to avoid costly investments on new tools and services;
- As digital CAD drawings were available on most of the owned facilities, the team could easily use baseline CAD files in Revit and place required asset components (Revit Families) on symbols shown on drawings; and
- The current COBie tools only generated a fixed set of common attributes for all model categories, but UW asset categories had many custom-defined attributes in addition to the common attributes (Table 2). Revit facilitated populating a different set of attributes for each asset category.

The rebaselining process, included creating asset categories (Revit Families), adding parameters or specified attributes to each asset category, importing CAD files into Revit, creating walls and spaces, placing asset components on CAD drawings/spaces, and populating asset data. For populating asset data, the team members used design and construction documents (e.g., drawings, manuals) for extracting baseline asset data and typing them in asset instances placed on the drawings.

The analysis of existing workflows showed that operation and maintenance staff needed to know the asset category each model component belonged to and where the assets were physically located. Staff typically did not need model components with high Levels of Development (LOD)[5] or placed in three-dimensional models for asset management tasks. These observations and reflections on the initial implementation of the workflow led the authors to streamline the workflow to include only the assets and asset attributes of interest to the owner. First, the team decided to create all Revit families under the “Generic” category to avoid issues concerning object-based behavior of other categories, such as the requirement for components to be hosted in walls or ceilings. Generic components in Revit do not contain any special aggregation relationships with other pre-defined categories. Additionally, this decision made the modeling process more efficient because the staff did not need training on different commands and functions in Revit to model different systems and categories. Each asset type (Revit Families) behaved in the same way. In the refined workflow, all asset families that are relevant to asset management (195 asset categories) could be accessed from a single command and drop-down menu accessible through the Architecture tab, Component’s Properties pane (see Figure 2). Since high LOD geometries were not required for maintenance and repair work, the team decided to use a simple box and a generic model that functioned as a symbol for each asset (low LOD 3D geometry) in each family. All components were color-coded such that modelers could easily identify the asset class to which each component belonged, such as orange for Fire, purple for Mechanical, green for Electrical, and blue for Plumbing (see Figure 74).


*Figure 2: Accessing all created asset categories (Families) from a drop-down menu.*

5. BIMForum. (2015). *2015 Level of Development Specification: buildingSMART International*.


*Figure 3: Creating low LOD components with different colors, shapes, geomteries based on the preferences of rebaselining team members.*

In order to facilitate setting-up future rebaselining projects, the team created a Revit template in which all asset families and data fields were pre-loaded. Therefore, the asset management staff would not need to create these components for other projects. For new rebaselining projects, the staff could open the template, create building levels, import drawings to each level, place pre-loaded components, and populate asset data (Figure 4).

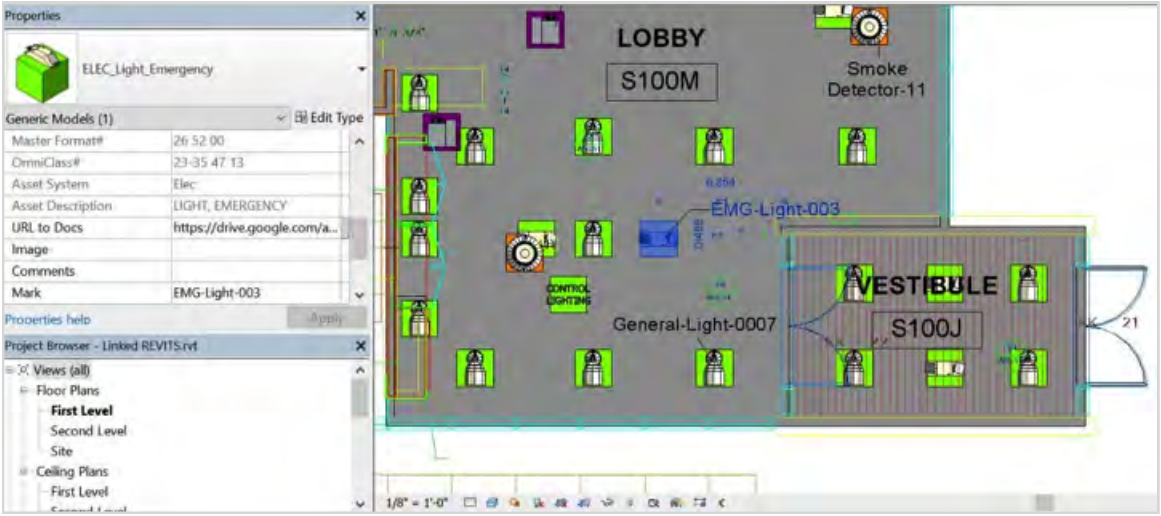



*Figure 4 – Asset data model created in Revit (asset instances + data attributes).*

After creating the asset data model, the team field-verified the model and asset attribute datasets in the selected facility. The team found that field verification was an indispensable part of the rebaselining workflow because the facility had numerous modifications that were not documented in drawings, project files, or asset data in the CMMS. Therefore, the team continued to update the data in the field in order to document the modifications. For example, many new lighting fixtures were added to the facility that were not shown in the drawings and some equipment had been removed or replaced. The team added comments to the asset components to document the absence of these assets in the drawings and keep a record of the field verification (e.g., comments such as 'asset removed' or 'new asset not documented in drawings').

The rebaselining effort demonstrated that the operations and maintenance staff may face difficulties in finding project documentation, such as manuals and drawings in project document archives. Therefore, field verification is essential in such cases to extract asset data from the information available on physical assets when possible. Nevertheless, some assets may also lack tags/plates, leading to incomplete or mismatched data attributes in the rebaselining effort regardless of field verification.

# 3 DETAILS OF REBASELINING WORKFLOW IN PROJECT 1: NAVISWORKS

This study showed that the rebaselining workflow should be implemented in the following four phases: (1) preparing technology enablers, (2) collecting asset data from existing documentations and archives, (3) on-site data verification, and (4) updating asset management databases. Each phase consists of a series of information inputs, processes, and data deliverables, which are discussed in detail in the following sections.

## 3.1. Phase 1: Preparing Technology Enablers

Phase 1 in the rebaselining workflow (Figure 5), which prepares technology enablers, involves setting up asset categories and attributes in the software selected for rebaselining. As rebaselining requires the latest version of asset data specifications, this phase must include coordination among all asset management service providers to update and verify asset data requirements. Next, a template file should be created in the rebaselining software to retain asset categories and attributes for the anticipated rebaselining efforts. Creating such a template requires different tasks and considerations in different rebaselining software. In Navisworks, these include generating the list of asset categories in a conventional XML Quantification Catalog, and adding the asset attributes to the default XML Configuration Catalog in Navisworks installed files.

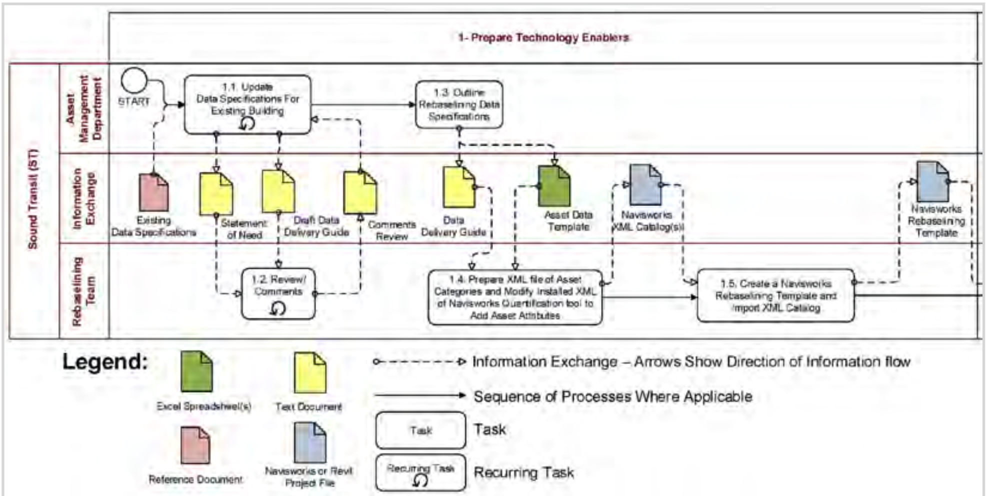
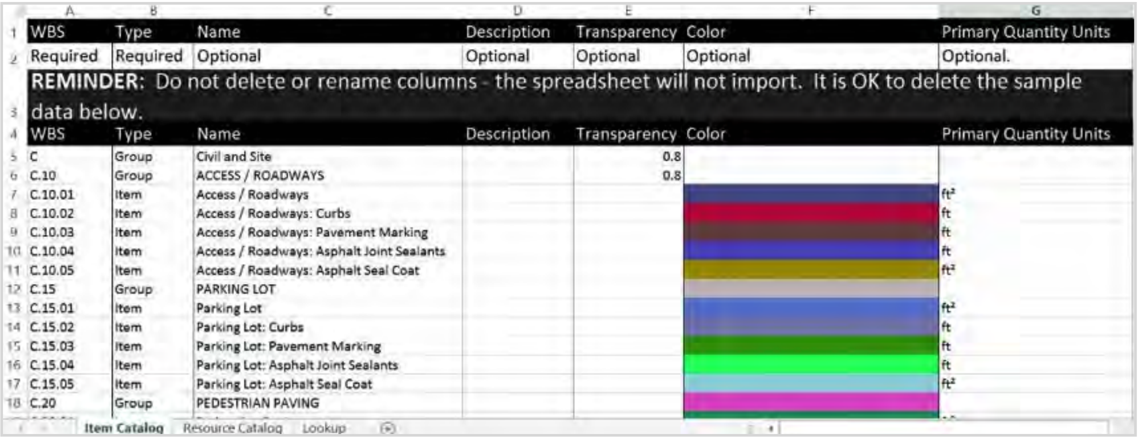



*Figure 5 – Phase 1 in the rebaselining workflow: Preparing technology enablers.*

### 3.1.1. Creating XML Quantification Catalog for Asset Categories

The asset data delivery guide (see Figure 5) should serve as the basis for creating XML Quantification Catalogs for rebaselining in Navisworks. Autodesk has developed a MS Excel spreadsheet tool[6] for creating XML Quantification catalogs for Navisworks. Asset systems and asset categories listed in the asset data delivery guide should be added to this Excel-based tool to form a work breakdown structure (WBS). Asset systems and asset categories can be color-coded and the required units of measure can be adjusted in this tool (Figure 6).

- Copy or type asset systems and asset categories to Name column in Autodesk Quantification Catalog Creation Template (Figure 6).
- Adjust the hierarchy of systems and their categories in WBS and Type column in the spreadsheet (the type of asset systems is "Group" and the type of asset categories is "Item").
- For exporting the spreadsheet to a XML Quantification catalog, in MS Excel, from ADD-INS Tab, choose Export Catalog (Figure 7).



| | A | B | C | D | E | F | G |
|---|---|---|---|---|---|---|---|
| 1 | WBS | Type | Name | Description | Transparency | Color | Primary Quantity Units |
| 2 | Required | Required | Optional | Optional | Optional | Optional | Optional. |
| 3 | REMINDER: Do not delete or rename columns - the spreadsheet will not import. It is OK to delete the sample data below. | | | | | | |
| 4 | WBS | Type | Name | Description | Transparency | Color | Primary Quantity Units |
| 5 | C | Group | Civil and Site | | 0.8 | | |
| 6 | C.10 | Group | ACCESS / ROADWAYS | | 0.8 | | |
| 7 | C.10.01 | Item | Access / Roadways | | | | ft² |
| 8 | C.10.02 | Item | Access / Roadways: Curbs | | | | ft |
| 9 | C.10.03 | Item | Access / Roadways: Pavement Marking | | | | ft |
| 10 | C.10.04 | Item | Access / Roadways: Asphalt Joint Sealants | | | | ft |
| 11 | C.10.05 | Item | Access / Roadways: Asphalt Seal Coat | | | | ft² |
| 12 | C.15 | Group | PARKING LOT | | | | |
| 13 | C.15.01 | Item | Parking Lot | | | | ft² |
| 14 | C.15.02 | Item | Parking Lot: Curbs | | | | ft |
| 15 | C.15.03 | Item | Parking Lot: Pavement Marking | | | | ft |
| 16 | C.15.04 | Item | Parking Lot: Asphalt Joint Sealants | | | | ft |
| 17 | C.15.05 | Item | Parking Lot: Asphalt Seal Coat | | | | ft² |
| 18 | C.20 | Group | PEDESTRIAN PAVING | | | | |

Item Catalog | Resource Catalog | Lookup

*Figure 6- A snapshot from Autodesk Navisworks Quantification Catalog Creation template.*


*Figure 7 – Exporting the XML Quantification Catalog.*

6. Download the tool and its instructions from: https://knowledge.autodesk.com/support/navisworks-products/downloads/caas/downloads/content/autodesk-navisworks-quantification-catalog-creation-template-2014.html.

### 3.1.2. Modifying Default XML Configuration Catalog to Add Asset Attributes to Navisworks

After exporting the XML Quantification catalog from the Autodesk tool, the required attributes listed in the asset data delivery guide (Figure 5) must be manually added to the Navisworks XML Configuration Catalog. The configuration file is located in the default installation folder of Naviswork (C:\Program Files\Autodesk\Navisworks Manage 201X\Quantification\templates) or any folders that users have chosen for installing Navisworks. The file name is “TakeoffConfigurationTemplate.xml.” The users must save a back-up file in another location before proceeding to edit it.

- Open “TakeoffConfigurationTemplate.xml” in a text editor software (e.g., windows Notepad or Notepad++).

This process requires editing three sections in this XML file. First, users need to add the custom attributes as new columns to this file. Column definition modules look like the following:

```
<Column Name="Description1">
    <Type>String</Type>
    <Purpose>Input</Purpose>
    <Formula></Formula>
    <Value varies="1"></Value>
    <Units></Units>
</Column>
```

- Adjust Column Name in the column definition module (type a custom attribute like “Asset ID” or “Manufacturer”), and copy the module (all 7 lines) to the file right after one of the column definitions already available in the file (Figure 8).

TakeoffConfigurationTemplate.xml - Notepad

File Edit Format View Help

```
      <Type>String</Type>
      <Purpose>Input</Purpose>
      <Formula></Formula>
      <Value varies="1"></Value>
      <Units></Units>
    </Column>
    <Column Name="Description2">
      <Type>String</Type>
      <Purpose>Input</Purpose>
      <Formula></Formula>
      <Value varies="1"></Value>
      <Units></Units>
    </Column>
        <Column Name="FacilityArea">
      <Type>String</Type>
      <Purpose>Input</Purpose>
      <Formula></Formula>
      <Value varies="1"></Value>
      <Units></Units>
    </Column>
        <Column Name="AreaServed">
      <Type>String</Type>
      <Purpose>Input</Purpose>
      <Formula></Formula>
      <Value varies="1"></Value>
      <Units></Units>
    </Column>
```

*Figure 8- Adding attributes to the Navisworks Configuration file.*

In the configuration file, two lines must refer to each column definition module. Each line consists of the following code:

*<ColumnRef Name= "XXXXXXXX" />*

- The ColumnRef code should be copied and added under existing references, and the exact name of the columns created in the previous step must be used in between " " marks.


```
    <ColumnRef Name="Description2" />
        <ColumnRef Name="FacilityArea" />
        <ColumnRef Name="AreaServed" />
        <ColumnRef Name="Serial" />
        <ColumnRef Name="Model" />
        <ColumnRef Name="Manufacturer" />
        <ColumnRef Name="FieldTag" />
    <ColumnRef Name="Count">
      <Purpose>Calculation</Purpose>
    </ColumnRef>
    <ColumnRef Name="PrimaryQuantity">
      <Purpose>Calculation</Purpose>
    </ColumnRef>
  </Table>

  <Table Name="StepResource">
    <ColumnRef Name="Count" />
    <ColumnRef Name="PrimaryQuantity" />
  </Table>

  <Table Name="ObjectStep">
    <ColumnRef Name="Object" />
  </Table>

  <Table Name="Step">
  </Table>

  <Table Name="Object">
    <ColumnRef Name="Object" />
    <ColumnRef Name="Description1" />
    <ColumnRef Name="Description2" />
        <ColumnRef Name="FacilityArea" />
        <ColumnRef Name="AreaServed" />
        <ColumnRef Name="Serial" />
```

*Figure 9 – Adjusting the references in the configuration file.*

- Save the file with the same name and file format (TakeoffConfigurationTemplate.xml).

The users need to make sure they do not overwrite the back-up file. If the users have worked with a XML Quantification Catalog (of asset categories) previously exported from Navisworks without having the new attributes in the XML Configuration Catalog, they need to delete a section of the XML catalog file in order to make it compatible with new settings.

- Open the file in Notepad++, use click on – icon to wrap <Catalog> and <ConfigFile> sections.
- Delete <Configfile> from the file and save the document (Figure 10).
- Import the catalog file to Navisworks without any issue as it reads new configurations from the XML Configuration file.

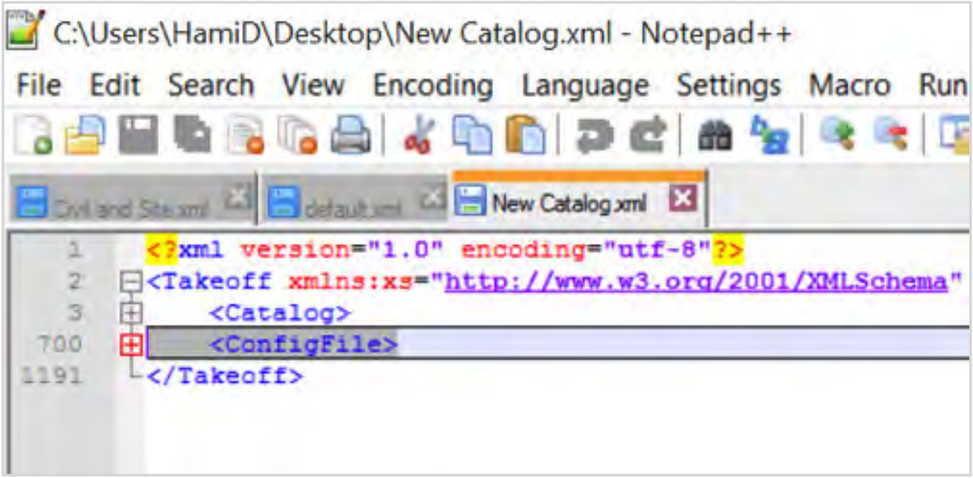



*Figure 10 – Fixing the existing XML Configuration Catalog (delete <ConfigFile>).*

### 3.1.3. Creating a Navisworks Rebaselining Template File

- Open Navisworks, and from the application menu, create a new project. This will be our Navisworks Rebaselining Template File (Figure 11).

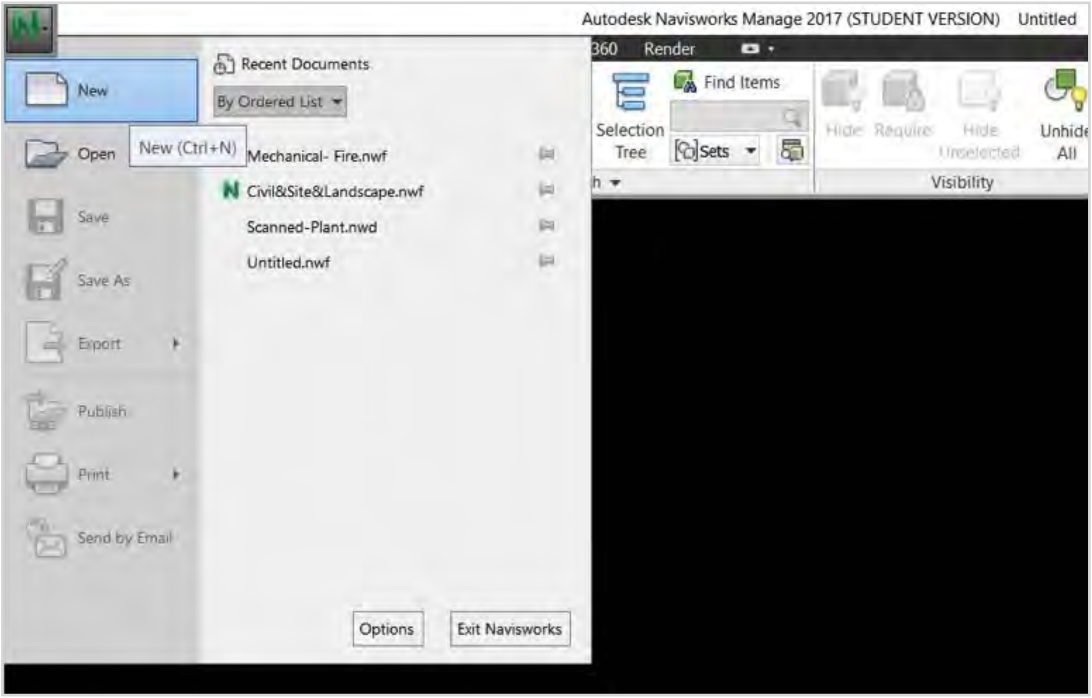



*Figure 11 – Creating a new Navisworks project.*

• Append all relevant drawings/documents (PDF or CAD files) to the template file using the Append command in Home tab (Figure 12). This process can be repeated if there are multiple documents/file available or new documents are generated for the project later in the process.

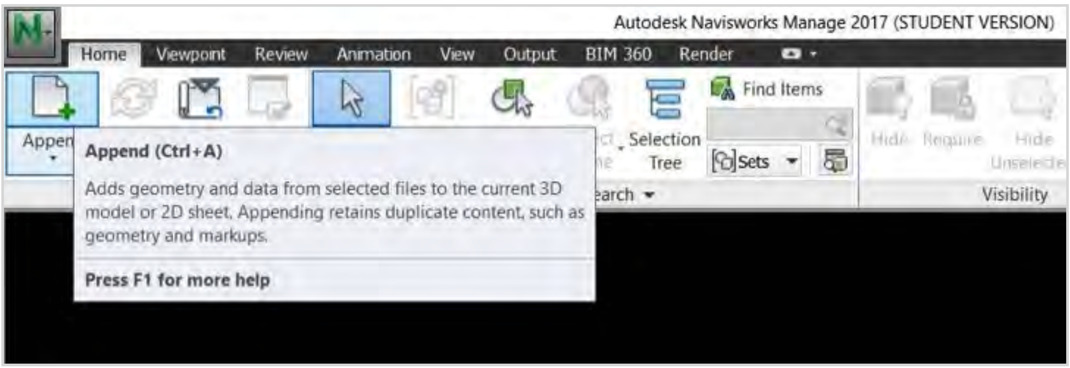



*Figure 12 – Append command in Home tab.*

• The appended documents can be explored through Sheet Browser window. To activate Sheet Browser, go to View tab > Windows command > enable Sheet Browser (Figure 13). In Sheet Browser, different sheets/pages of each appended document can be viewed by double clicking on the sheet/page name or number.

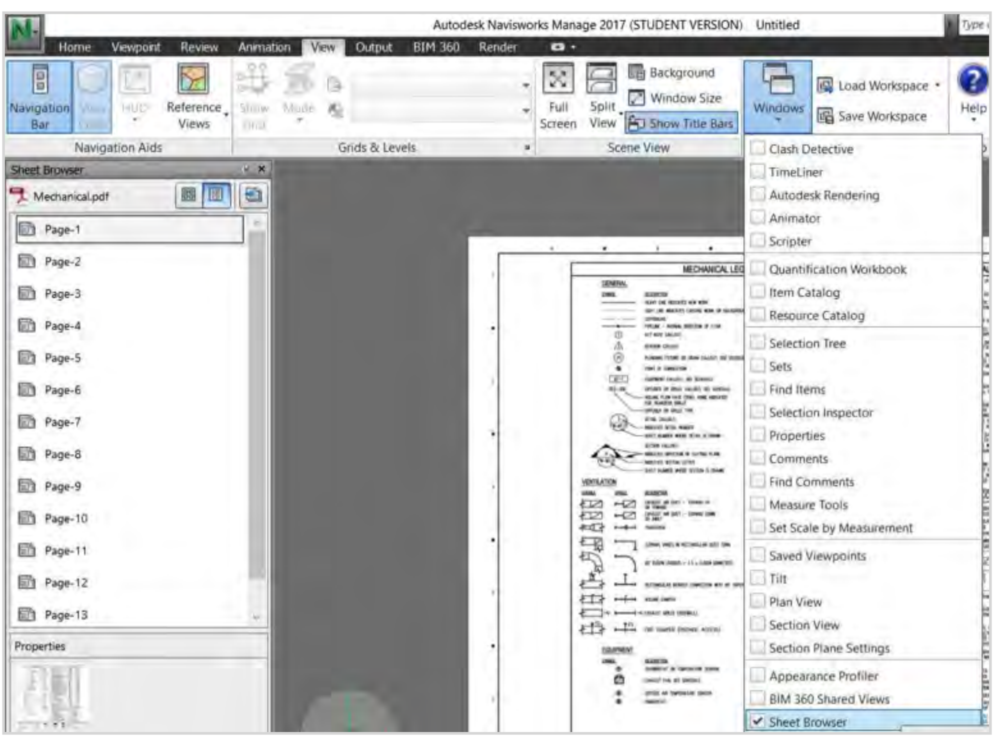



*Figure 13 – Enabling Sheet Browser window in Navisworks interface.*

After appending drawings/documents, import the XML Quantification Catalog created in the Autodesk Navisworks Quantification Catalog Creation tool. Navisworks will use the asset categories of the imported XML Quantification Catalog and the asset attributes of the XML Configuration Catalog.

- Enable the Quantification tool in Home tab > Tools panel (Figure 14) to open Quantification Workbook and click Project Setup in this window (Figure 15).

- In the pop-up window (Quantification Setup Wizard), users must select “Browse to a catalog” and click Browse to locate the previously saved XML Quantification Catalog and open it (Figure 16).


*Figure 14 – Enabling Quantification tool.*

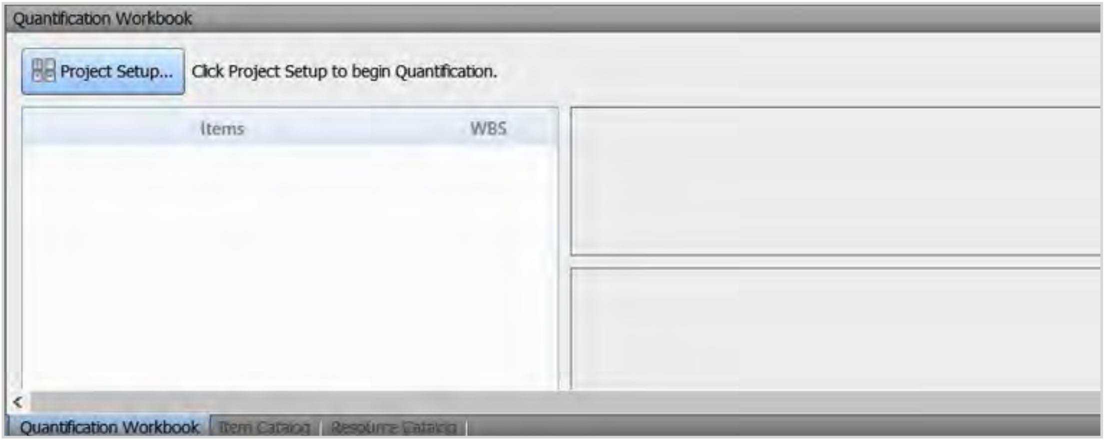



*Figure 15 – Setting up a quantification project.*

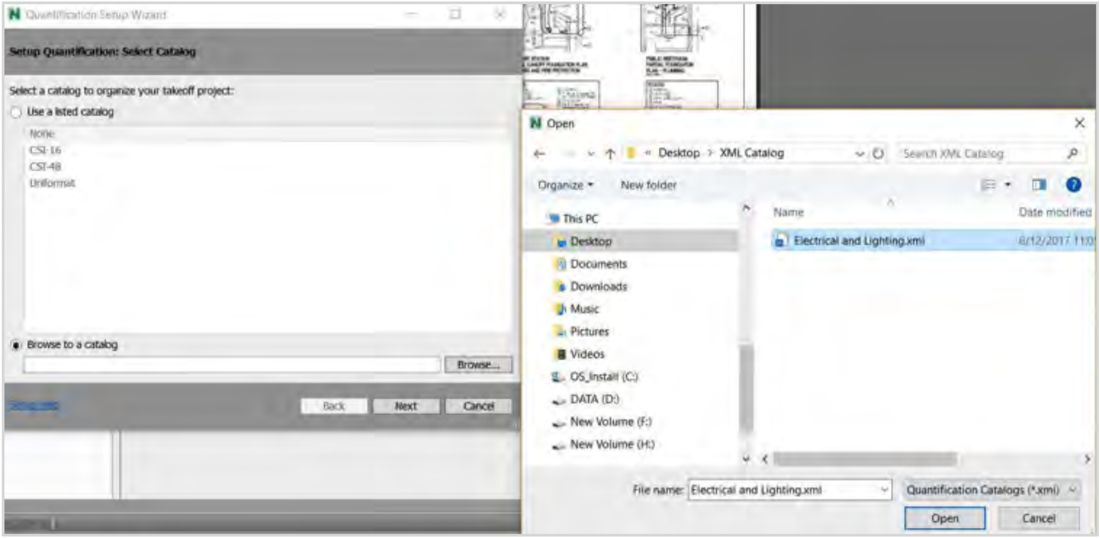



*Figure 16 – Importing XML Quantification Catalog into Navisworks Quantification tool (Browse to catalog).*

• After opening the XML Quantification Catalog, the Quantification Workbook's left pane shows the asset systems and the asset categories per asset data delivery guide under the Items column (Figure 17).

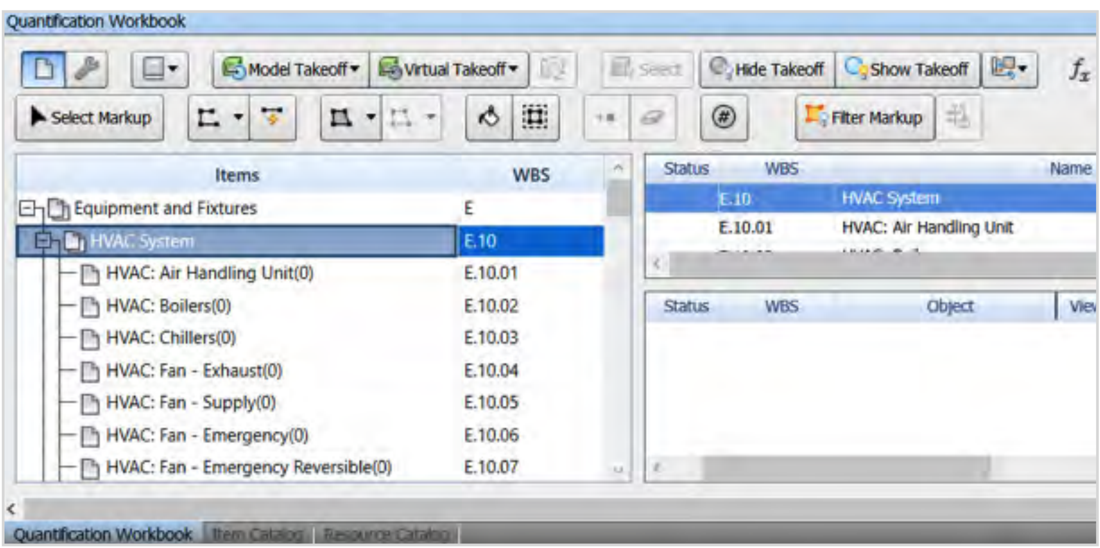



*Figure 17 – The imported list of asset systems and asset categories.*

• Save the rebaselining project template file using the application menu's Save As command. NWF filetype must be selected for saving the file (Figure 18) since NWF files create links to separated pages/sheets of imported documents and this makes the file size smaller and faster to process. NWD filetype makes all pages/sheets internal to NWD files and this usually makes the file size huge and slow to process.

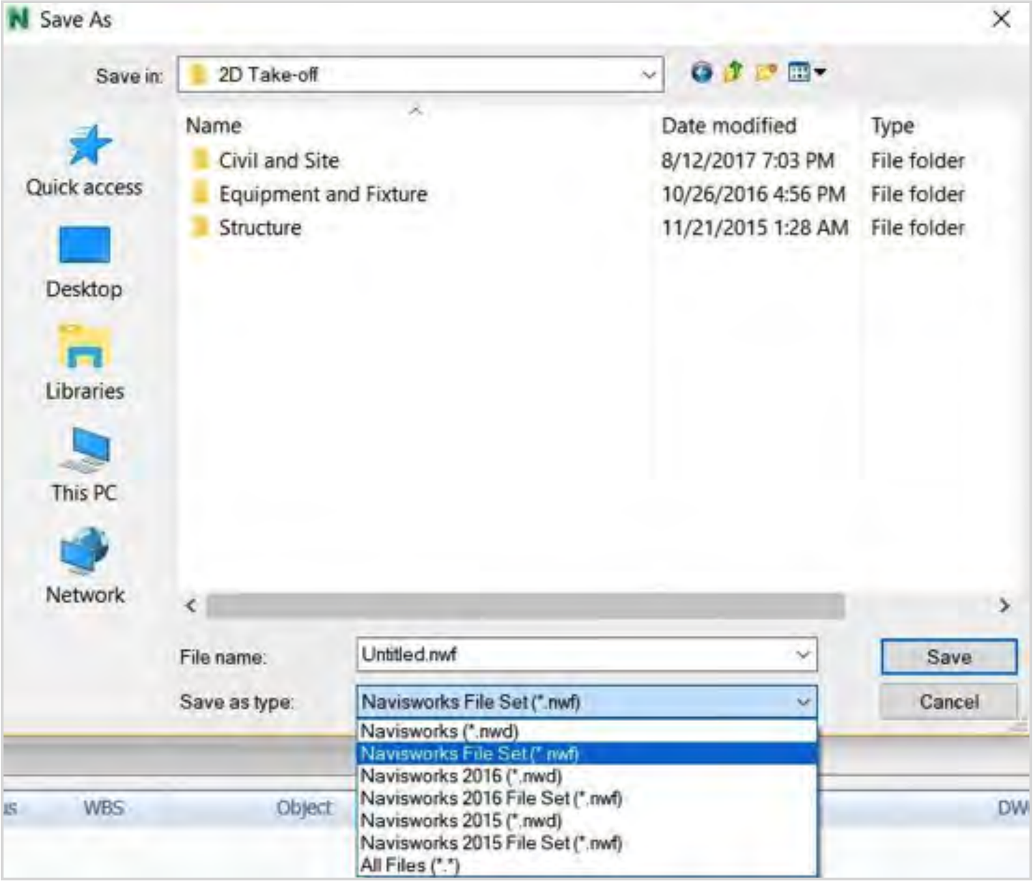



*Figure 18 – Saving the Navisworks project as NWF.*

## 3.2. Phase 2: Collecting Asset Data from Existing Documentation Archives

In the phase 2 (Figure 19), users must make sure all project drawings/ documentations are appended into the rebaselining software so the users can place asset instances or mark-ups as an overlay. The users should also extract asset data from different sources (e.g., manuals, drawings, schedules, brochures) and aggregate them in data fields that are linked to the mark-ups. The rebaselining workflow is limited by data collection issues such as lost or missing documents, unavailable data (not commissioned in project documents), and conflicting data (e.g., conflicts between O&M manuals and schedules on drawings). These issues must be documented in a rebaselining log file in order to be coordinated with asset management staff and reconciled during field verification if possible. Facilities management staff may have personal knowledge of assets or keep personal files in their daily practice that can also be used to refine the asset dataset during rebaselining. After this coordination process, asset data models can be accordingly updated.

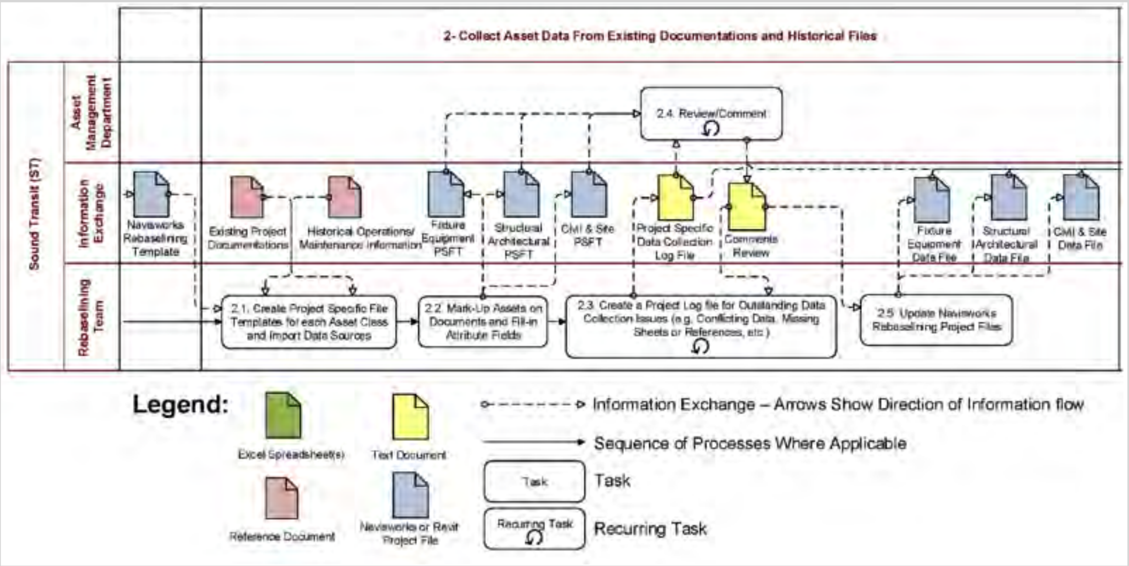



*Figure 19 – Phase 2 in the rebaselining workflow: collecting asset data.*

### 3.2.1. Scaling Drawings

To extract reliable quantities from drawings, the imported documentations/ drawings must be scaled to 1:1.

- Open a sheet/page from Sheet Browser and enable Set Scale by Measurement window from View tab > Windows command.
- In Set Scale by Measurement window, click New Measurement, and draw a line on the drawing to specify its real-world length/ dimension in Enter Value pop-up window (Figure 20; drawing on existing dimensions, scales, or using the distance between grid lines are suitable for this purpose since grids are visible in most drawings).
- Click the Finish in Set Scale by Measurement window to apply the new scale to the drawing sheet (Figure 21). This process must be repeated for each sheet/page in Sheet Browser to make sure the quantities are accurate.

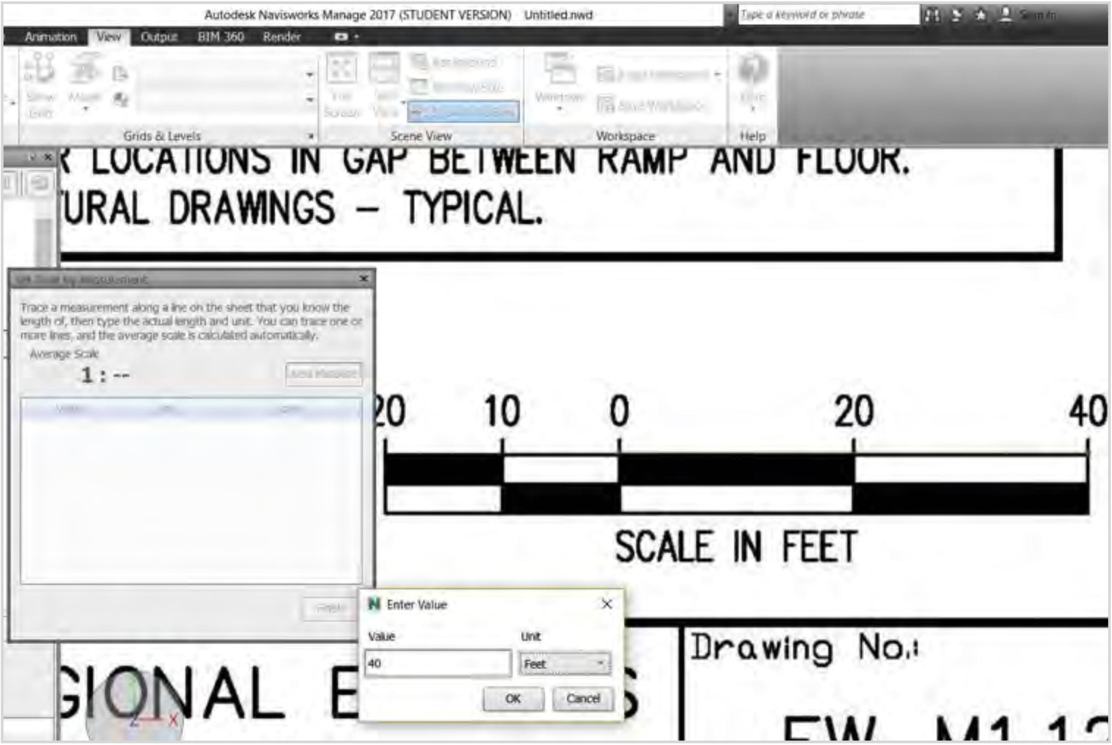



*Figure 20 – Set scale of drawings by measurement (a line is drawn from 0 to 40 and its length is specified in Enter Value window).*


*Figure 21 - Click Finish to apply the new scale to the drawing sheet.*

## 3.2.2. Visualizing Asset Instances with Mark-ups

• For placing mark-ups (asset instances) on drawings, make sure the Navisworks interface shows Sheet Browser and Quantification Workbook windows (Figure 22).These windows can be enabled through the View tab > Windows command.

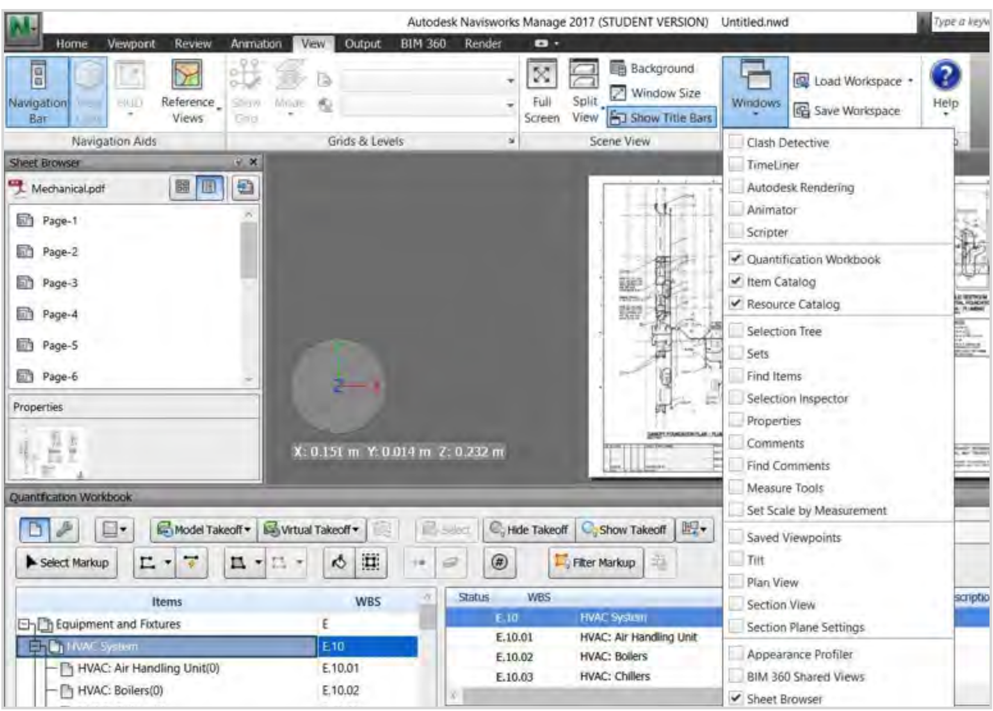



*Figure 22 – Enabling Sheet Browser and Quantification Workbook.*

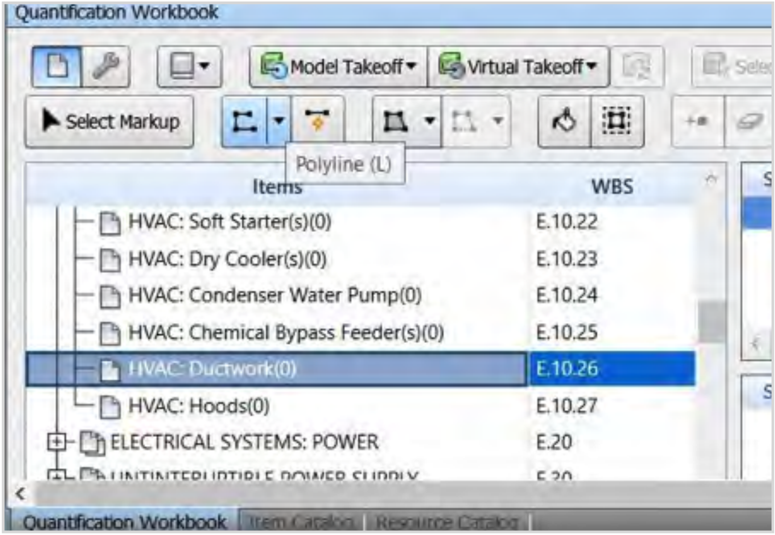



*Figure 23 – Selecting an asset category (HVAC Ductwork) and the relevant mark-up tool (polyline).*

• Next, in Quantification Workbook, select one asset category at a time and enable a relevant markup tool (Figure 23). In Quantification Workbook, there are different mark-up tools, such as line, polyline, area, rectangle area, and count (Figure 24). These can be directly placed/drawn on documents/drawings.


*Figure 24 – Mark-up tools in Quantification Workbook*

• After placing a mark-up for an asset instance, a new row will be added for that mark-up in the data table in Quantification Workbook, categorized under the selected asset category. The asset instances area is listed under the Object column in the data worksheet (Figure 25, section 3).

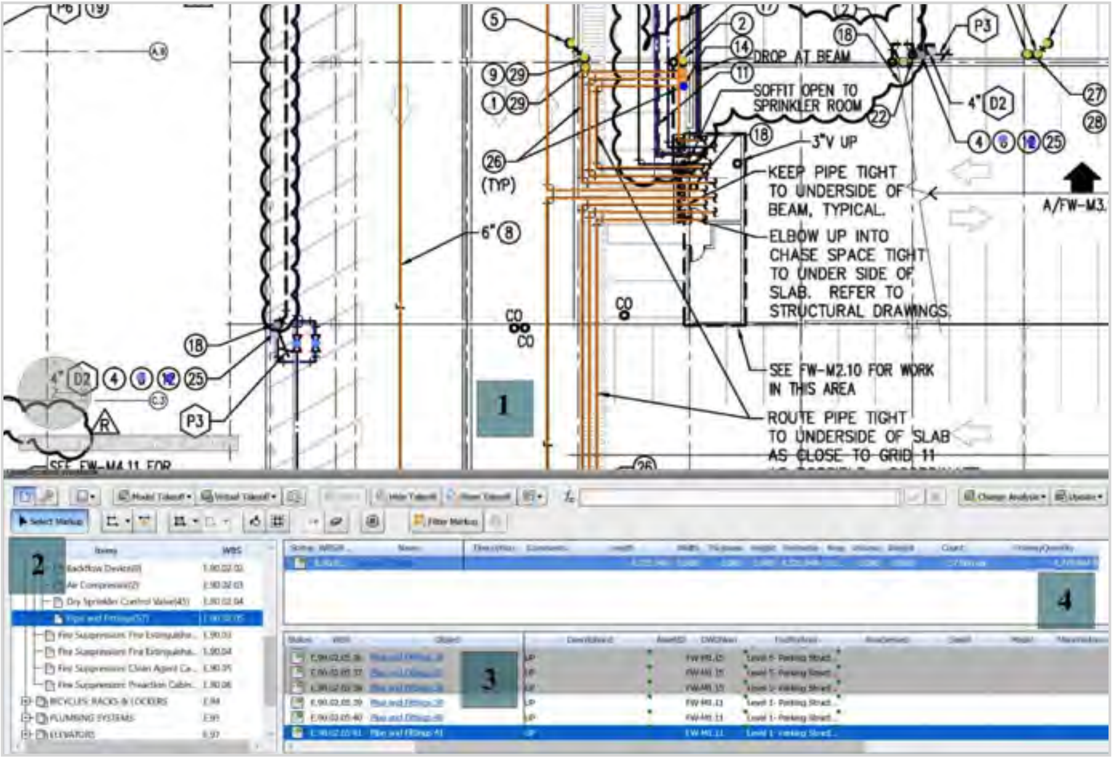



*Figure 25 – 1: Mark-ups on drawings; 2: Asset categories; 3: Asset instances and attributes in data worksheet; 4: Total quantity for the selected asset category.*

## 3.2.3. Aggregating Asset Data Attributes in the Data Worksheet

Asset attributes are also listed in the data worksheet as column names (Figure 25, section 3). For each asset instance, its attributes can be manually added to the relevant row and column. Quantities data (e.g., count, length, area) are automatically generated in the data worksheet based the size/ count of mark-ups. The total quantity for each asset category is shown in the Total Quantity section in the Quantification Workbook (Figure 25, section 4). For risers and vertical linear segments, 'count' mark-ups must be placed and quantities should be manually typed-in to override existing data (Figure 26).

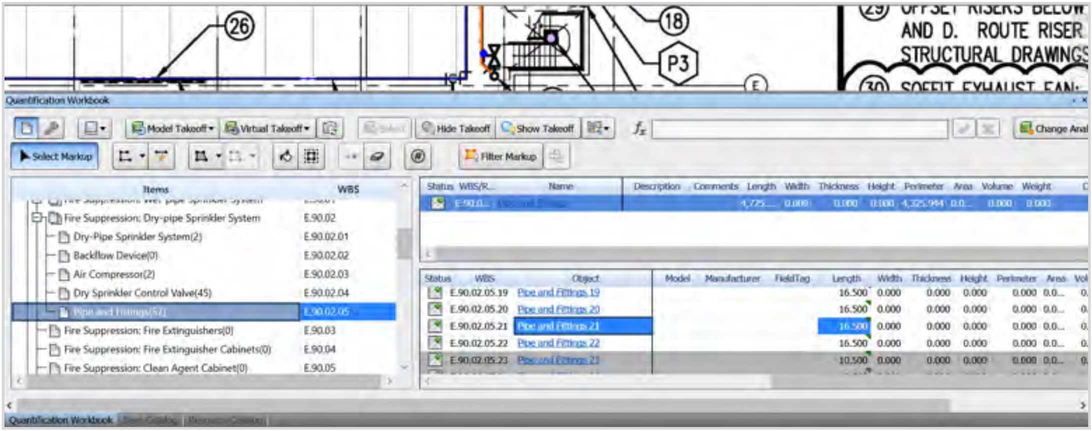



*Figure 26- Overriding the length parameter for a riser.*

As mentioned above, mark-ups and data fields are linked together. Therefore, if a user wants to use Asset ID to find an asset among the many similar assets on a drawing, the user can sort the table based on the Asset ID column, right click on the object cell of the wanted asset, and click Select Take-off's Markups. Navisworks will automatically show the mark-up in the viewport if the mark-up is in the opened sheet (Figure 27).

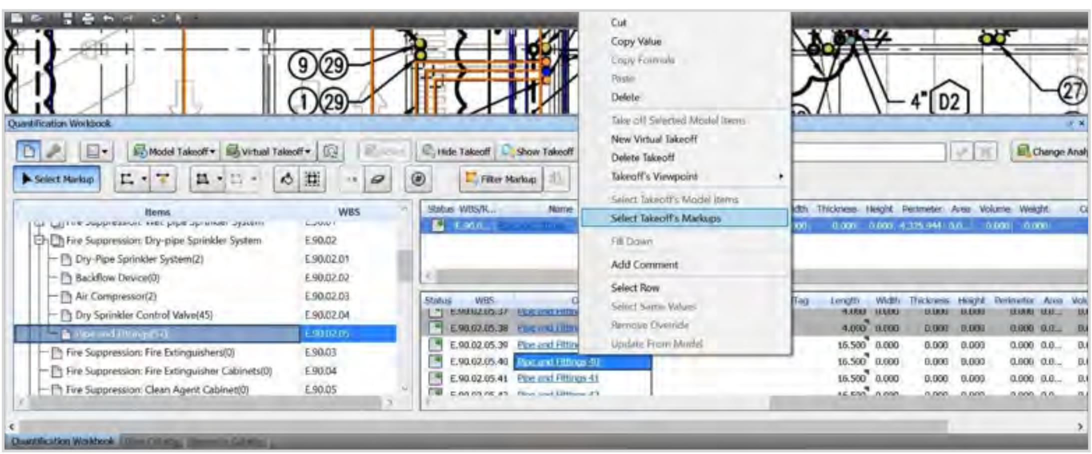



*Figure 27- Finding asset mark-ups by selecting data fields.*

Data rows become gray when the mark-ups they represent are not on the currently opened sheet in the viewport (Figure 28) and the linked mark-ups are unable to be selected. For this reason, we suggest that the DWGNUM (Drawing Number) be added to the required attributes in the XML Configuration Catalogs to streamline the process of finding mark-ups and assets on different sheets and pages of appended documents (Figure 29).

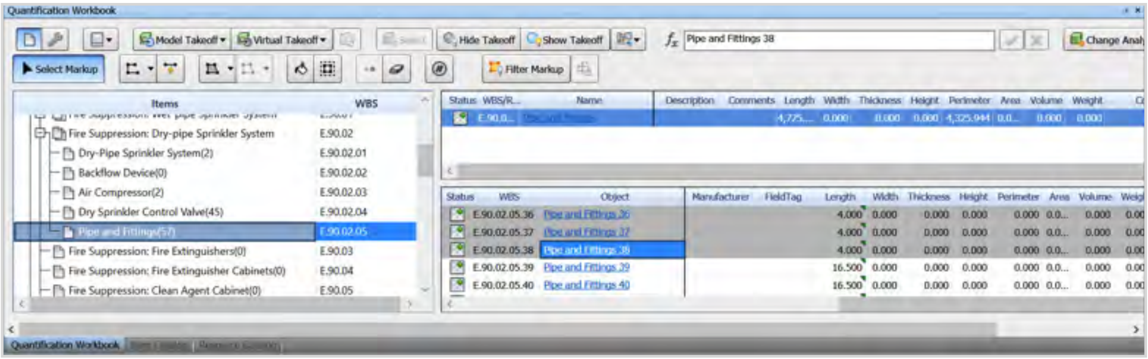



*Figure 28- Grayed Data Rows.*

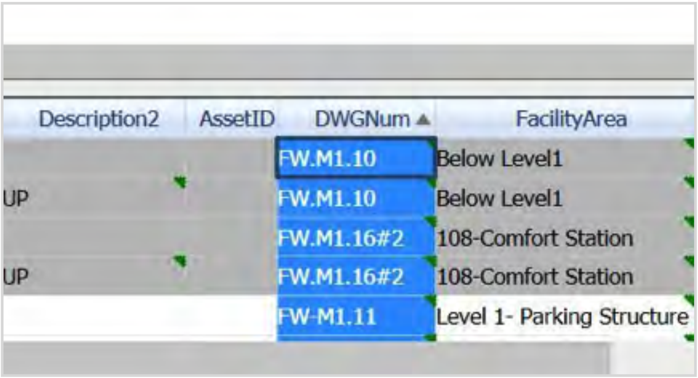



*Figure 29- Adding drawing numbers to data attributes can streamline the processes.*

## 3.3. Phase 3: On-site Data Verification

Phase 3 in the rebaselining workflow (Figure 30) addresses the field verification of asset data and, if necessary, updating asset data models or physical asset tags/plates. These processes are essential in the rebaselining workflow—and in asset management in general—because existing facilities often have undocumented asset modifications that cause interruptions in asset operation, maintenance, and renewal. Not only can this rebaselining process facilitate identifying and documenting these modifications in asset data models, but it is also an opportunity to compare documented data against physical asset tags to find data that are missing from manuals or drawings. For this reason, implementing this phase can significantly improve the accuracy and reliability of asset data models.

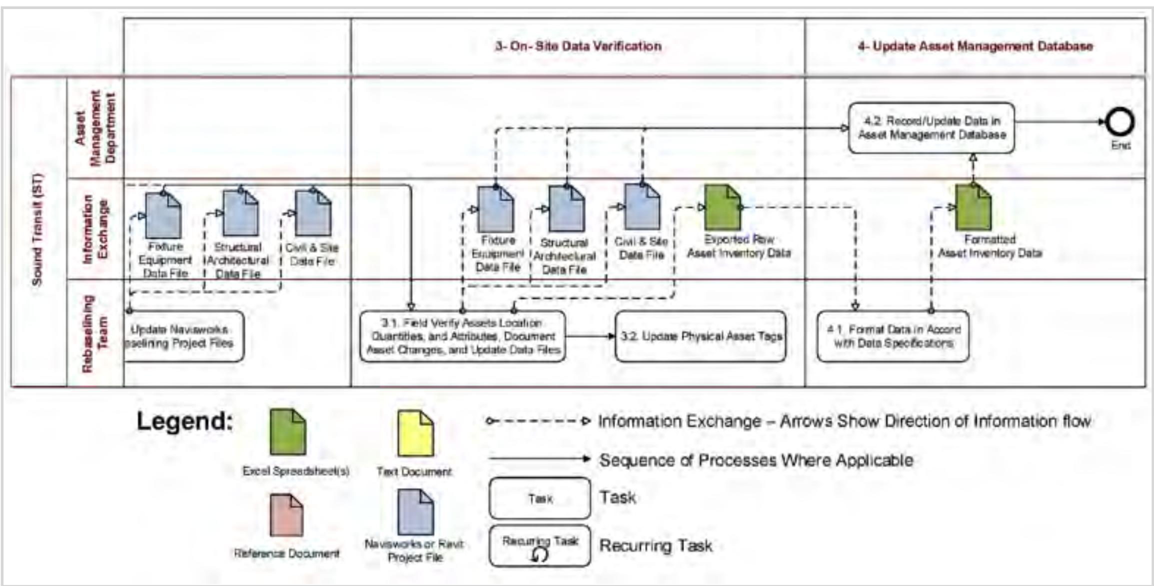



*Figure 30 – Phase 3 and Phase 4 in the rebaselining workflow: On-site data verification and updating asset management database.*

In Navisworks, asset data in data worksheets can be easily modified or regenerated based on the field verification process.

## 3.4. Phase 4: Update Asset Management Database

Navisworks can export data from the verified asset data to raw data files in order to be prepared in the format required for CMMSs. As the owners or asset management staff may require asset inventory data in a specific format (e.g., COBie standard spreadsheet), phase 4 is included for transforming raw data files to different formats (Figure 30). Those responsible for implementation must decide if the formatted asset inventory data must be machine-readable (for CMMS), human readable, or both because certain CMMSs may require specific data structures to support interoperable data transfer.

### 3.4.1. Exporting Asset Data to MS Excel Files

To export asset data to a MS Excel file, click the Import/Export menu in the Quantification Workbook and select Export Quantities to Excel (Figure 31). For partial data extraction, choose Export Selected Quantities to Excel after selecting data rows in data worksheets. Excel files are flexible in terms of reformatting and editing data and preparing them for interoperable data exchange.

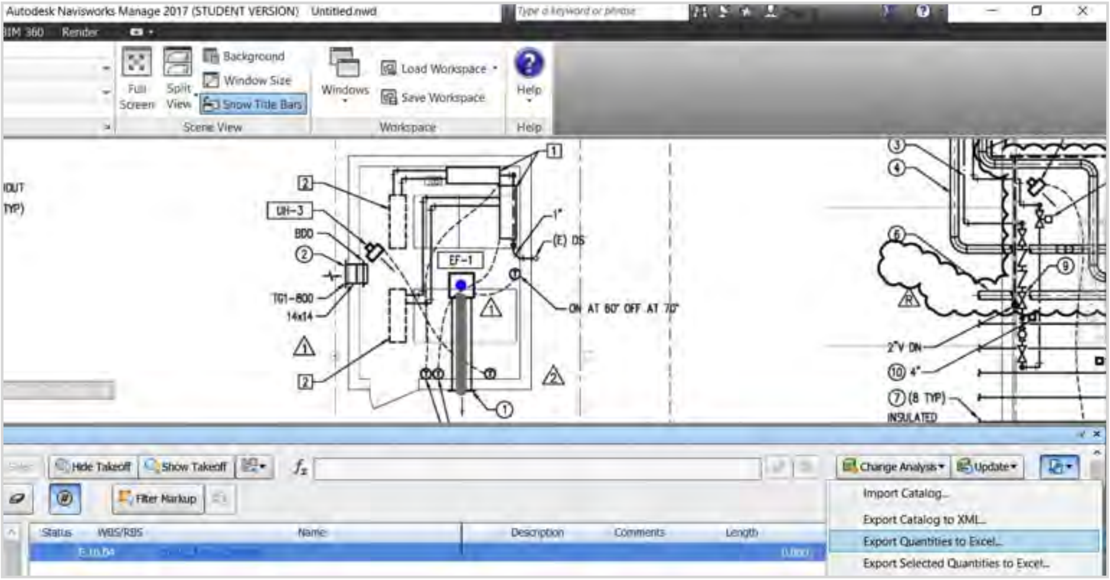



*Figure 31 – Exporting asset data/quantities to MS Excel file format.*

## 3.5. Modifying and Exporting Asset Categories List in Navisworks

The rebaselining workflow should allow for updates to the list of asset systems, asset categories, and asset attributes in the rebaselining database (Navisworks in this case). Asset systems and asset categories can be adjusted in the Item Catalog window, which can be enabled from the View tab > Windows command (Figure 32).

- For creating new asset systems, use New Group; for creating new asset categories, use New Items. These items and groups can also be deleted or renamed in this section.

- In Item Catalog, users can set colors and transparencies for mark-ups of each asset category (Figure 32).

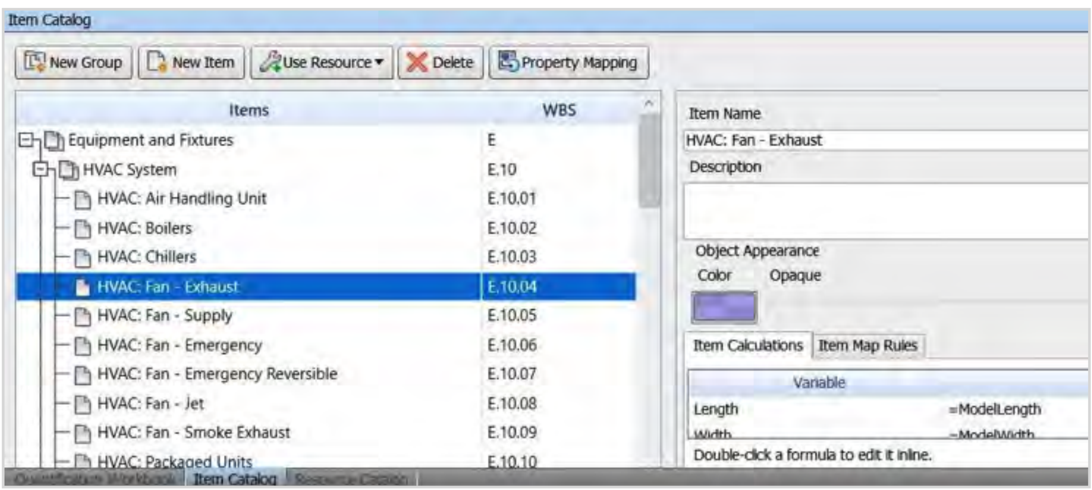



*Figure 32 – Using Item Catalog for modifying asset systems/categories.*

After updating the list of asset systems and asset categories, a new XML Quantification Catalog can be exported for creating a back-up or for use in other rebaselining projects.

- Choose Export Quantification Catalog in the Import/Export menu of the Quantification Workbook (Figure 31). For modifying asset attributes, follow the processes shown in section 2.1.2 to modify the XML Configuration Catalog.

# 4 REBASELINING WORKFLOW IN PROJECT 2: REVIT

This study showed that the rebaselining workflow should be implemented in four phases, including (1) preparing technology enablers, (2) collecting asset data from existing documentations and archives, (3) on-site data verification, and (4) updating asset management databases. Each phase consisted of a series of information inputs, processes, and data deliverables, which are discussed in details in the following sections.

## 4.1. Phase 1: Preparing Technology Enablers

Phase 1 in the rebaselining workflow (Figure 1), which prepares technology enablers, involves setting up asset categories and attributes in the software selected for rebaselining. As rebaselining requires the latest version of asset data specifications, this phase must include coordination among all asset management service providers to update and verify asset data requirements. Next, a template file should be created in the rebaselining software to retain asset categories and attributes for the anticipated rebaselining efforts. Creating such a template requires different tasks and considerations in different rebaselining software.

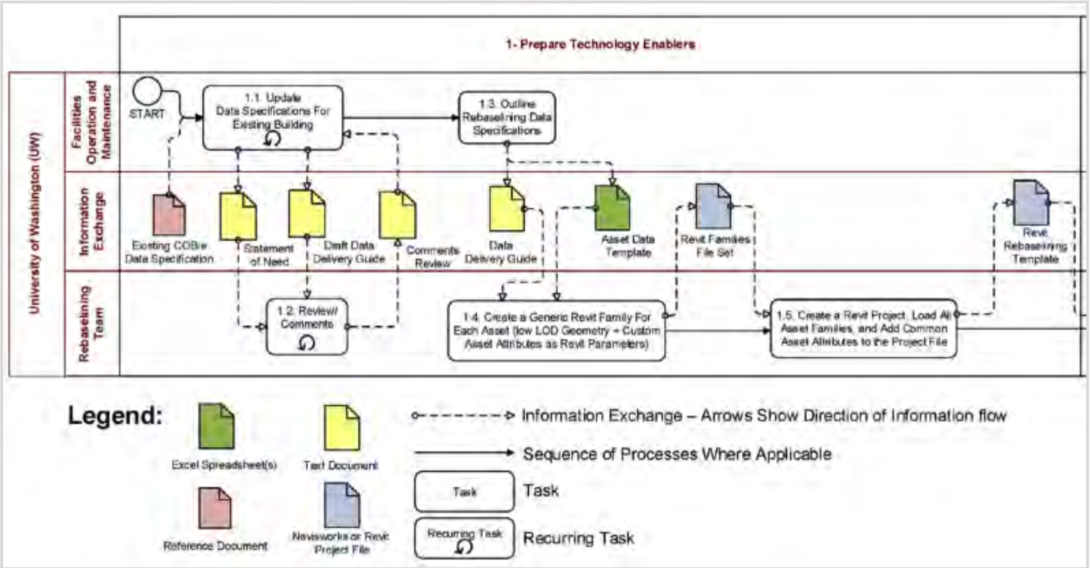



*Figure 33 – Phase 1 in the rebaselining workflow: Preparing technology enablers.*

### 4.1.1. Setting up Asset Attributes in Revit

In the first step, the users need to create a Shared Parameters file to archive all required asset attributes in a document that can be used in current and future projects. Shared parameters are attributes that can be assigned to asset categories and/or asset instances. Since we want to create asset categories in the Family Editor environment of Revit, we need to create Shared Parameters in the same environment.

- Open Revit. From the Application Menu, choose New > Family (Figure 34).
- From the browser, among Family Templates, choose Generic Model.rft (Figure 35).
- Revit opens the Family Editor environment. In the Family Editor environment, choose Manage tab > Settings panel > Shared Parameters command (Figure 36).

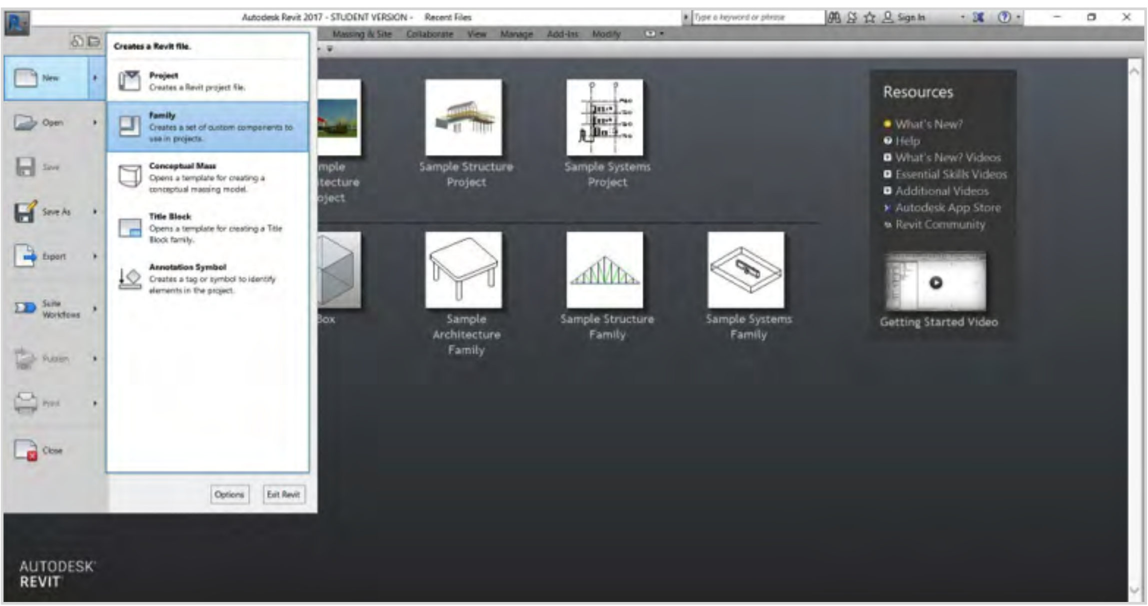



*Figure 34 – Open Revit and create a new family.*

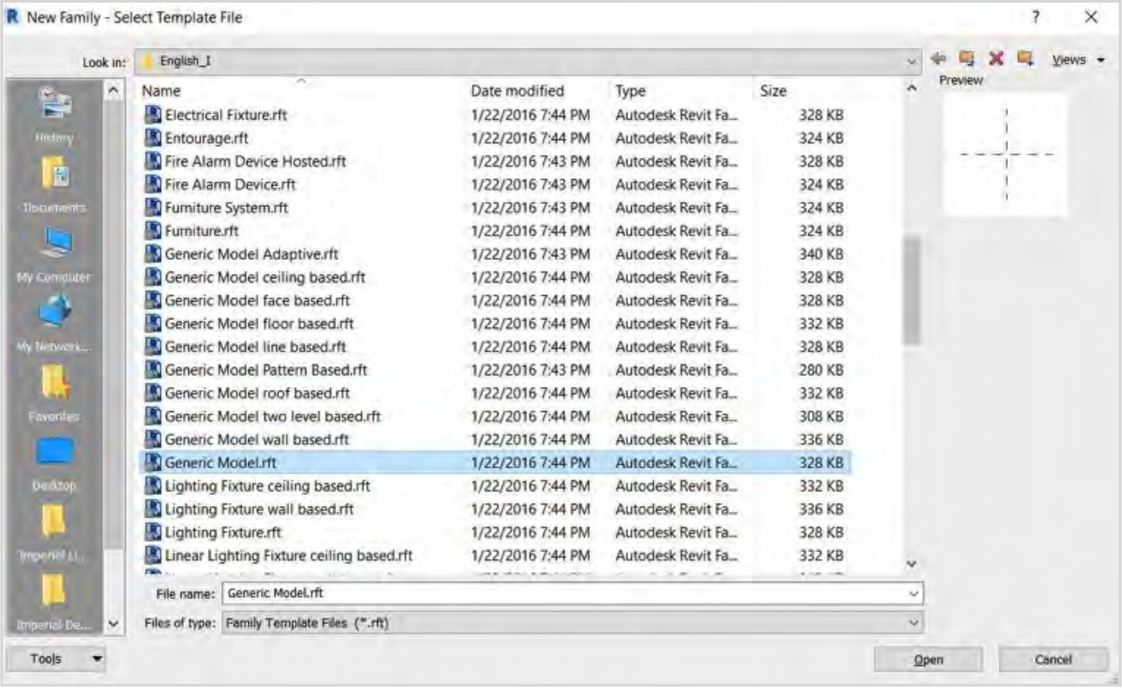



*Figure 35 – Open Generic Model.rft as the required family template.*

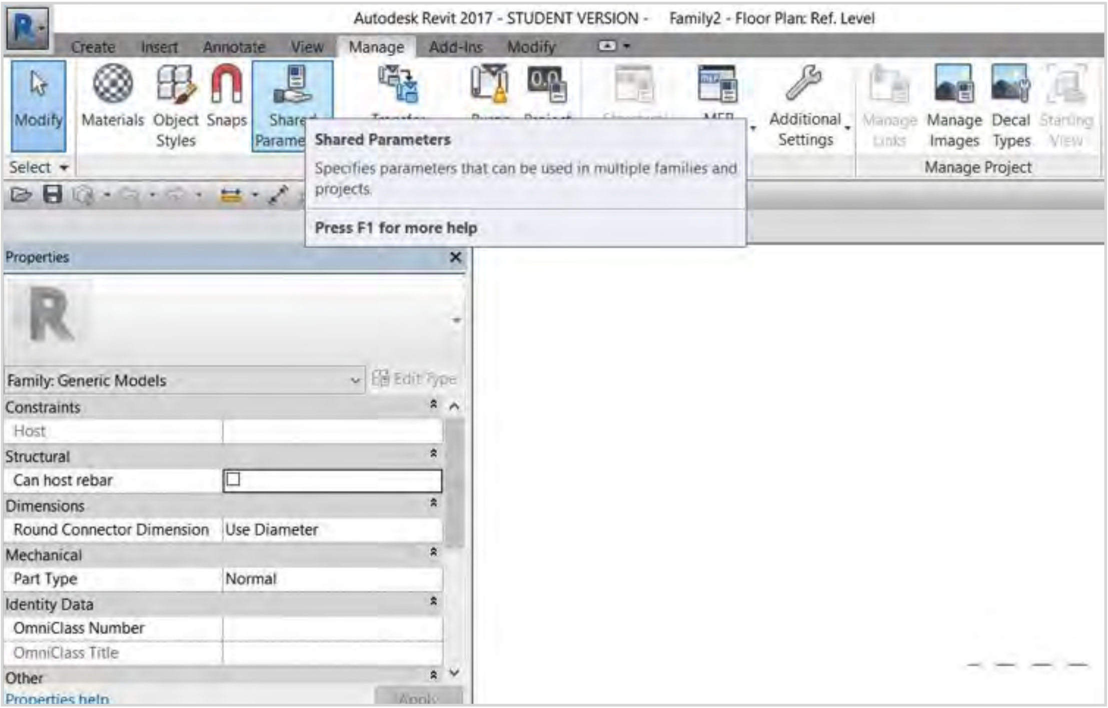



*Figure 36 – In the Family Editor Environment, from the Manage tab, open Shared Parameters.*

• In the Edit Shared Parameters window, click Create, and save the Shared Parameter text file in a folder (Figure 37 and Figure 38).

• In the Edit Shared Parameters window, create a new Parameter Group to begin categorizing asset attributes (Figure 39).

• After creating Parameter Groups, select a group from the Parameter Group drop-down menu. From the Parameters section, click New to create a new Parameter (or asset attribute) and set its name, discipline, and type (Figure 40 and Figure 41).

• Figures 42 and 43 show the Parameter Groups and a set of Parameters created for the project.


*Figure 37 – Create a new Shared Parameter file.*

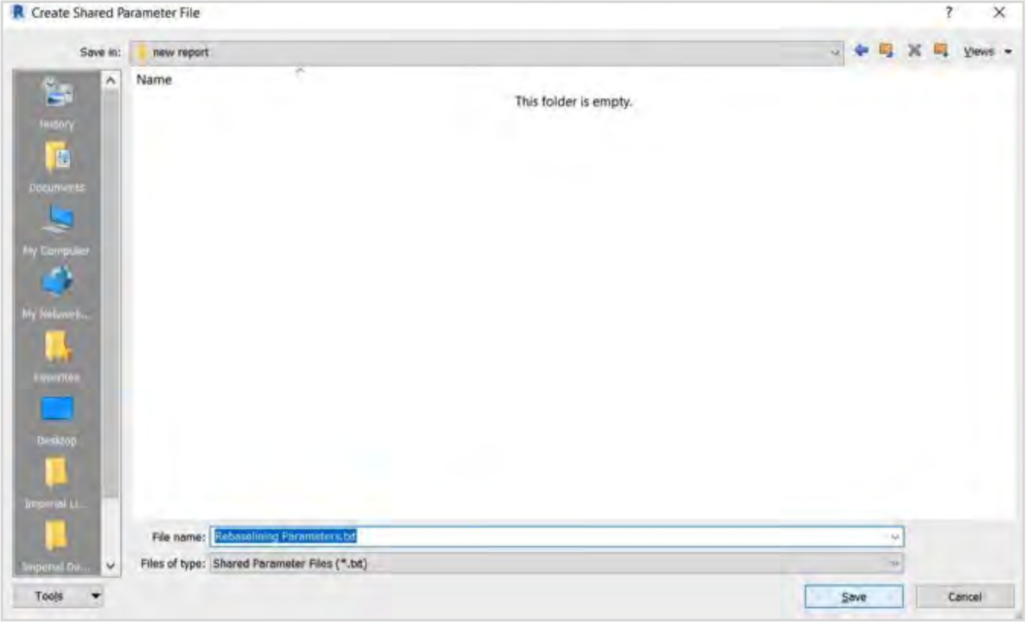



*Figure 38 – Save the Shared Parameter file.*

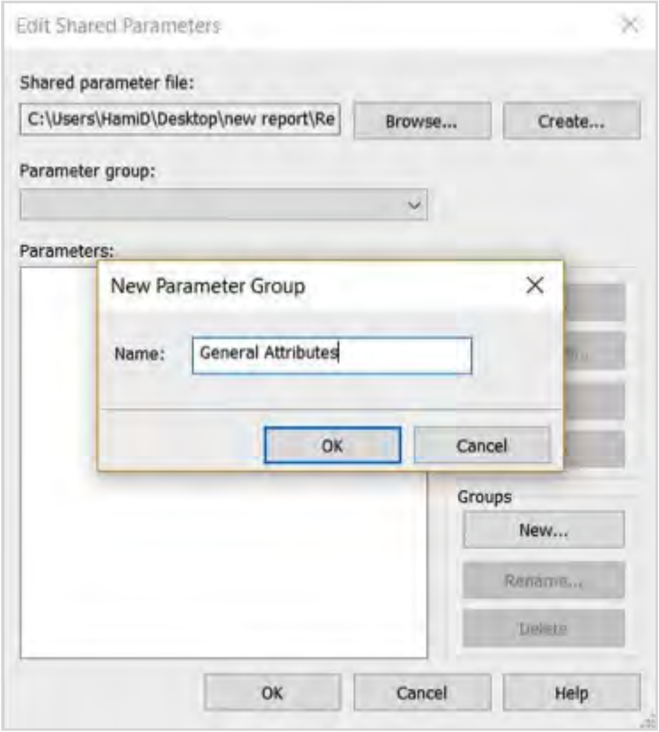



*Figure 39 – Creating a new Parameter Group.*


*Figure 40 – Create a New Parameter under a relevant Parameter Group.*

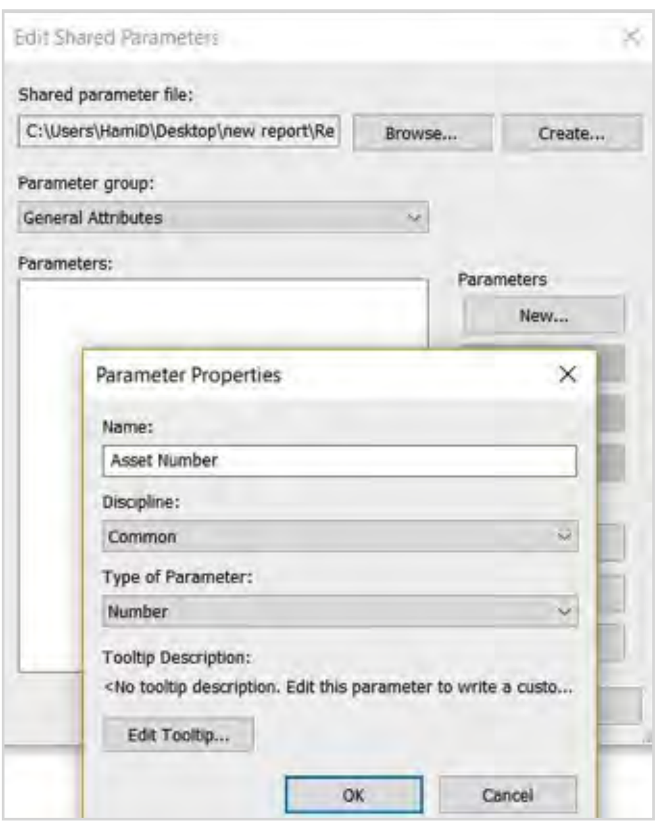



*Figure 41 – Set the name, discipline, and type of parameters.*

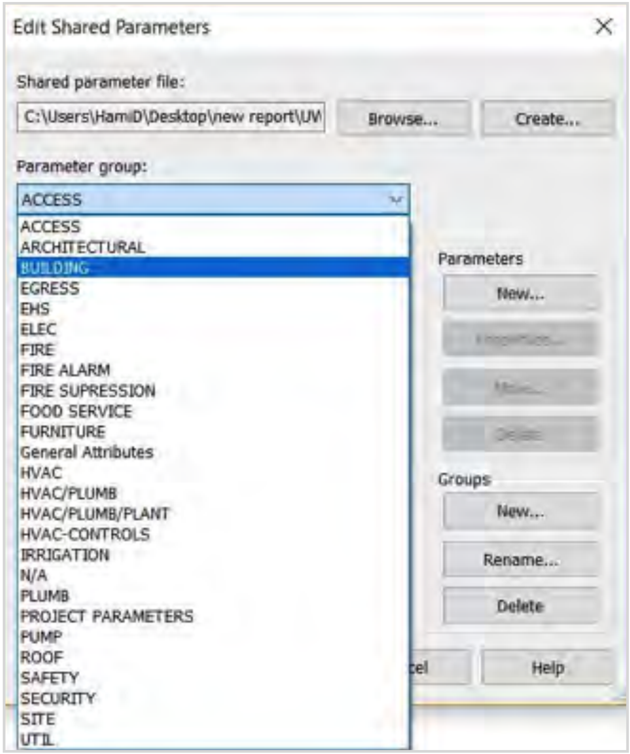



*Figure 42 – Parameter Groups set for the project.*


*Figure 43 – Sample parameters created under the Project Parameters group.*

The Parameters group and the parameters created in this process or added later will be automatically saved into the Shared Parameters text file created in the first step as long as the address bar in the Edit Shared Parameters window shows that file is opened. By clicking Browse, a previously saved Shared Parameters file can be loaded into Revit.

### 4.1.2. Creating Generic Asset Family

In this workflow, we create all Revit families under the “Generic Model” category to avoid issues concerning object-based behavior of built-in Revit categories (such as the requirement for components to be hosted in walls or ceilings). Generic components in Revit do not contain any special aggregation relationships with other pre-defined categories. Additionally, this decision made the modeling process more efficient because the staff did not need training on different commands and functions in Revit to model different systems and categories. From a geometric standpoint, each asset type (Revit Families) behaved in the same way, although their attributes might be different per the asset data delivery guide.

- Open a new Family file with the Generic Model.rft template (Figure 34 and Figure 35)

- On the Create tab, click Extrusion (Figure 44).

- On the left side of the interface in the Project Browser, you can see that Ref. Level under Floor Plans is activated, so we see a plan view of this Revit family.

- After enabling the Extrusion tool, under Modify tab > Draw panel, choose Rectangle tool (Figure 45).

- Draw a rectangle in the viewpoint by placing two opposite corners of a rectangle (Figure 46). After drawing the rectangle, click on the value shown next to the dimension lines and type a new value for the length of the corresponding edges (Figure 47).

- Under the Modify tab, click the green check mark to finish edit mode (Figure 48).

- Select the rectangle by clicking on its edge, and drag the rectangle toward the center of the green axes so that the center point in the rectangle matches the center point of the axes (Figure 49).

• Click Default 3D View to see the box (the extruded triangle) we just created (Figure 50).

• Select the box by clicking one of its edges. In the Properties window on the left, you can change the size of the extrusion to change the height of this box.

• From the application menu, save this family file in a folder and name it Generic_Asset.rfa (Figure 51).

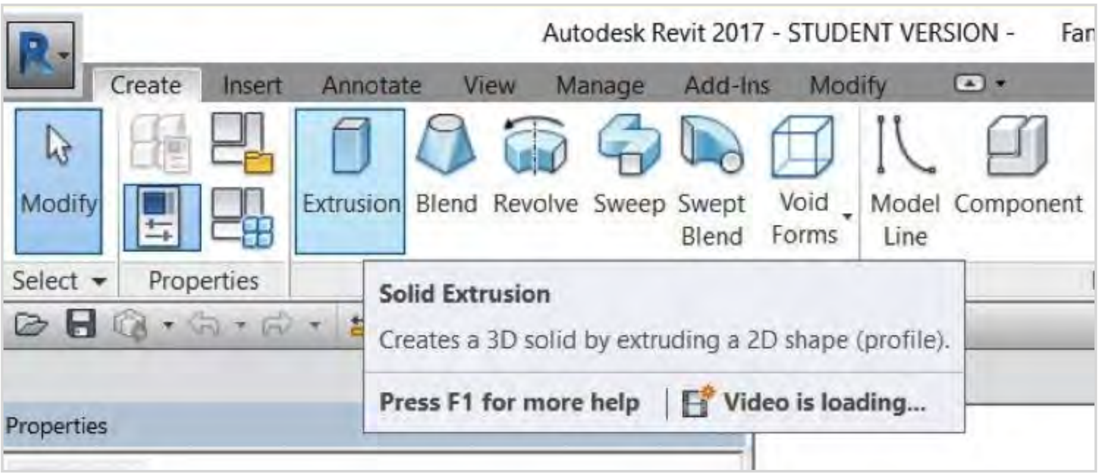



*Figure 44 – Enable Extrusion command.*

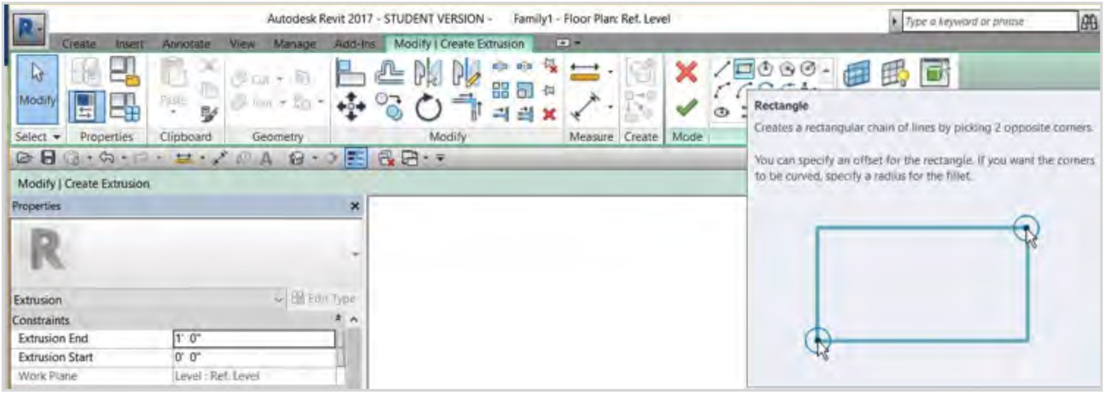



*Figure 45 – Choose rectangle among drawing tools.*

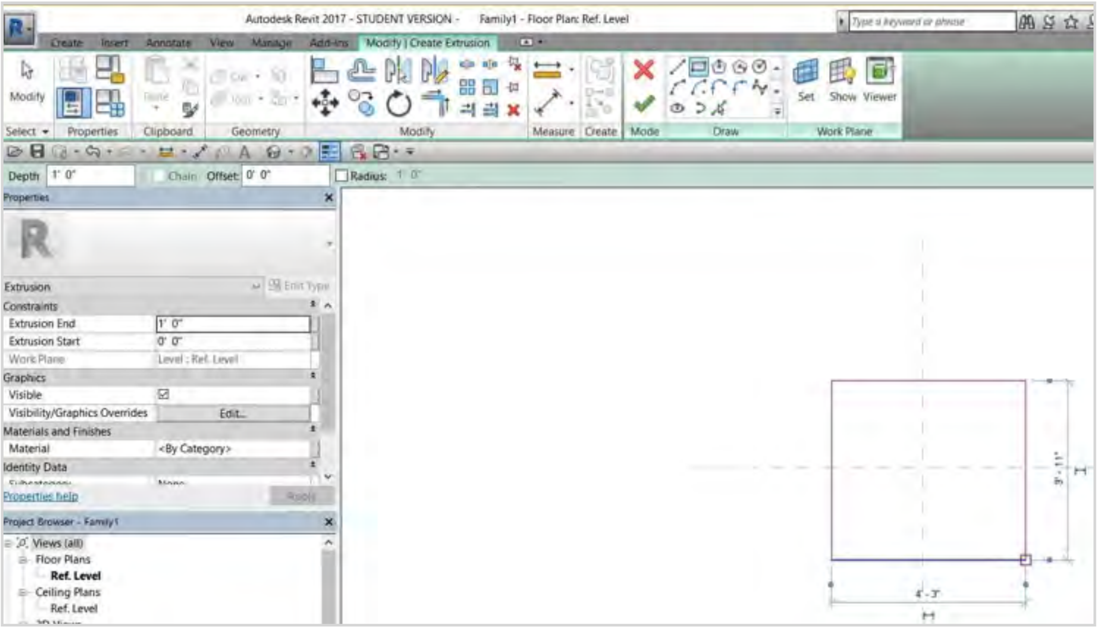



*Figure 46 – Draw a rectangle by placing corners.*

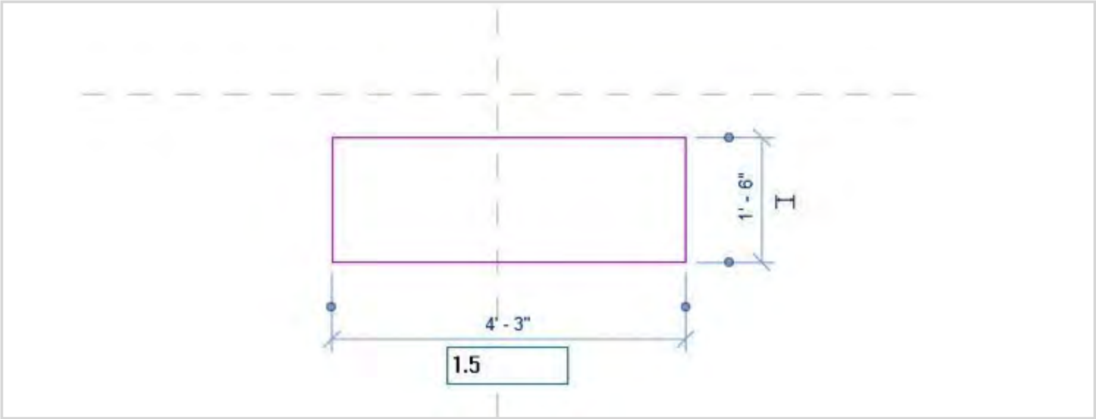



*Figure 47 – Set the length of edges by changing values on dimension lines.*

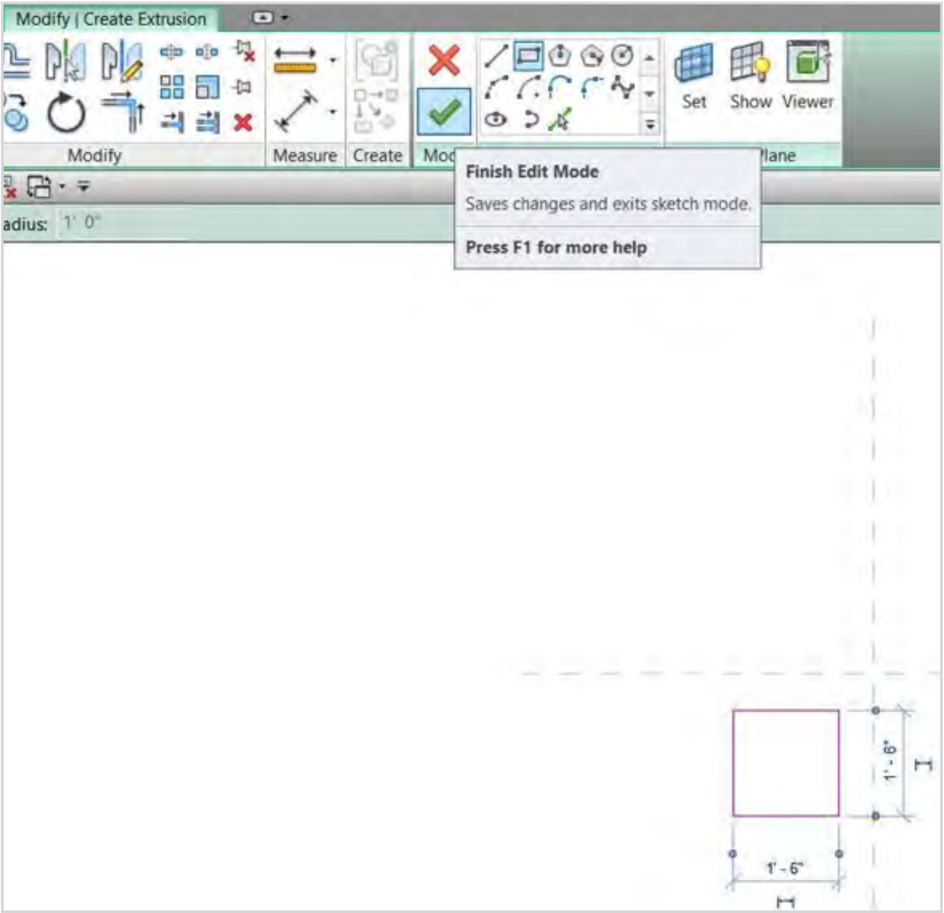



*Figure 48 – Finish Edit Mode.*

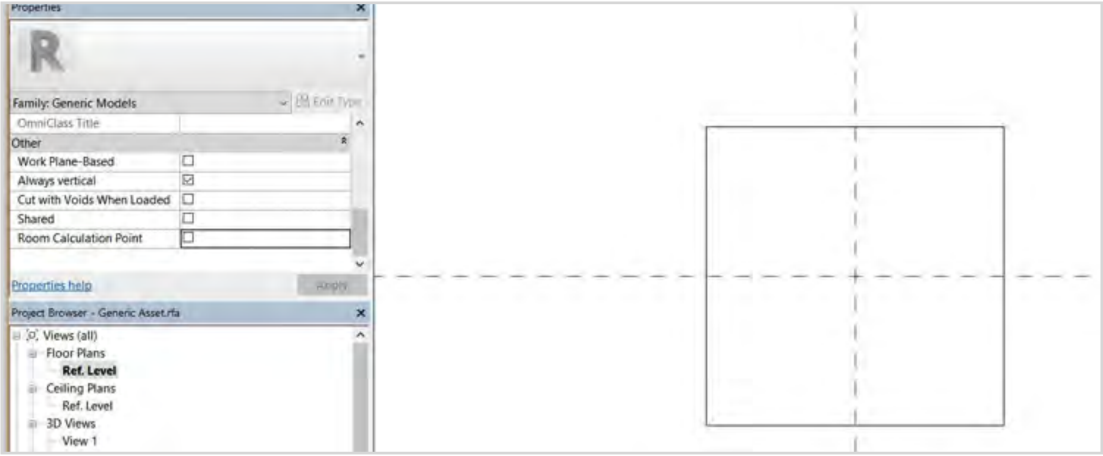



*Figure 49 – Moving the rectangle to the center point.*

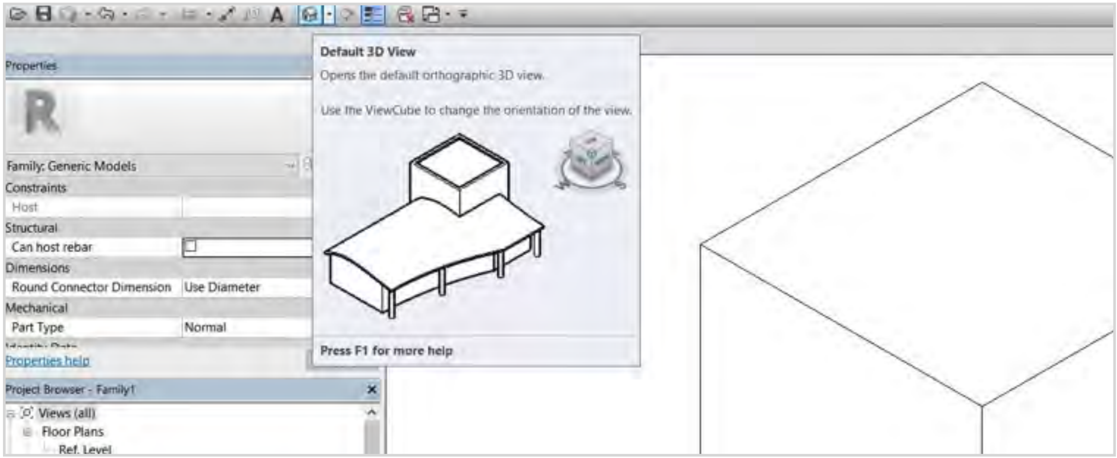



*Figure 50 – Activate Default 3D View.*

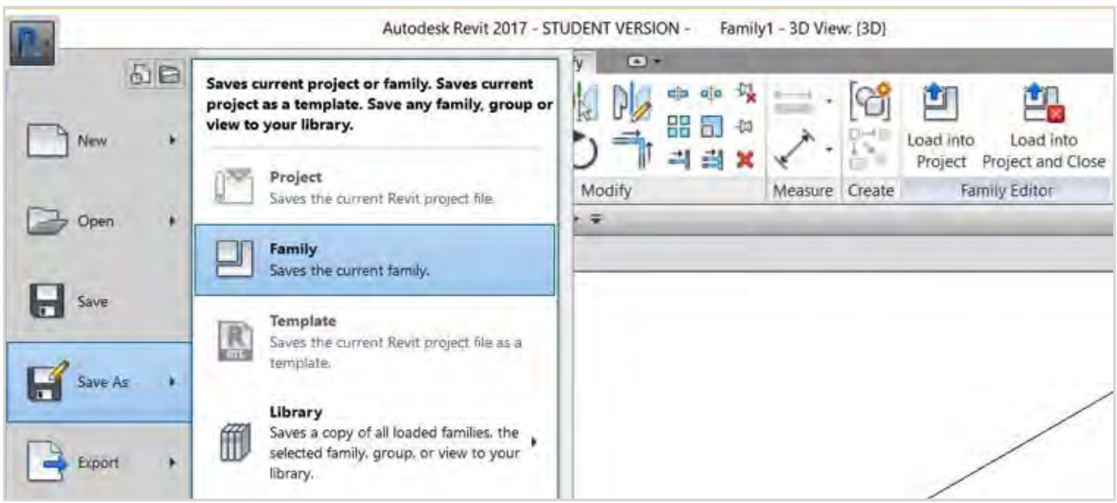



*Figure 51 – Save the Family file.*

In the next step, we want to add some attributes or parameters to the generic asset. The attributes we want to add in this step are asset attributes that all asset categories must individually have. These attributes include:

- AssetID
- Uniformat#
- Master Format#
- OmniClass#
- Asset System
- Asset Description

These parameters should be added to the generic asset family using the following process:

- From the Create tab's Properties panel, open Family Types (Figure 52).
- In the Family Types window, click New Parameter (Figure 53).
- In the Parameter Properties window, enable Shared Parameters and click Select. From the Shared Parameters window, and a relevant Parameter Group, find one of the required parameters and click OK (Figure 54).
- After selecting the parameter, in the Parameter Properties window, select Instance for Parameter type, and Group Parameter Under "Identity Data." Parameters that are added as instance parameters will have "(default)" in front of their parameter name in the Family Type window (Figure 55).
- Repeat the process for adding other parameters.
- Save the generic asset file.

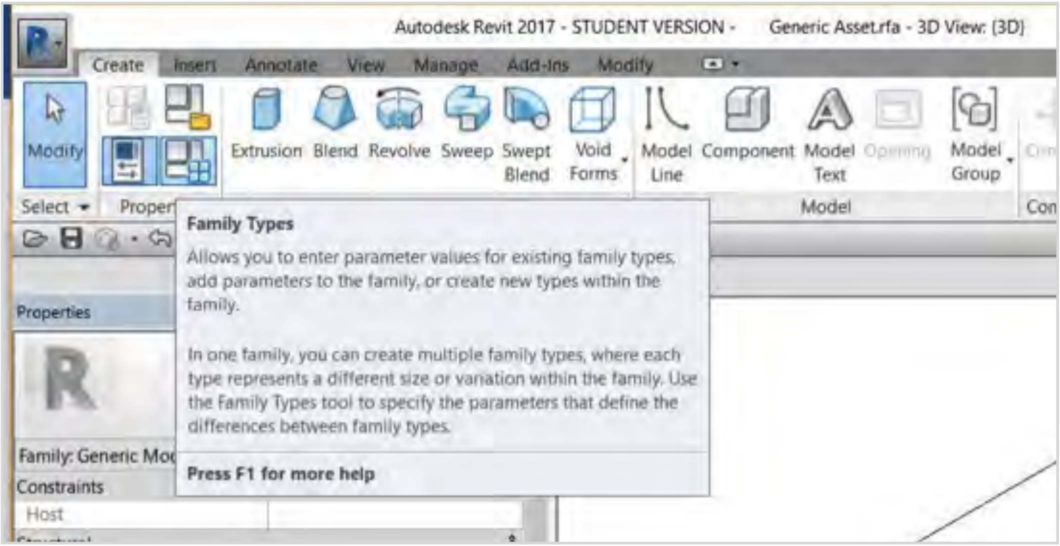



*Figure 52 – Open Family Types Window.*


*Figure 53 – Choose New Parameter from the lower toolbar.*

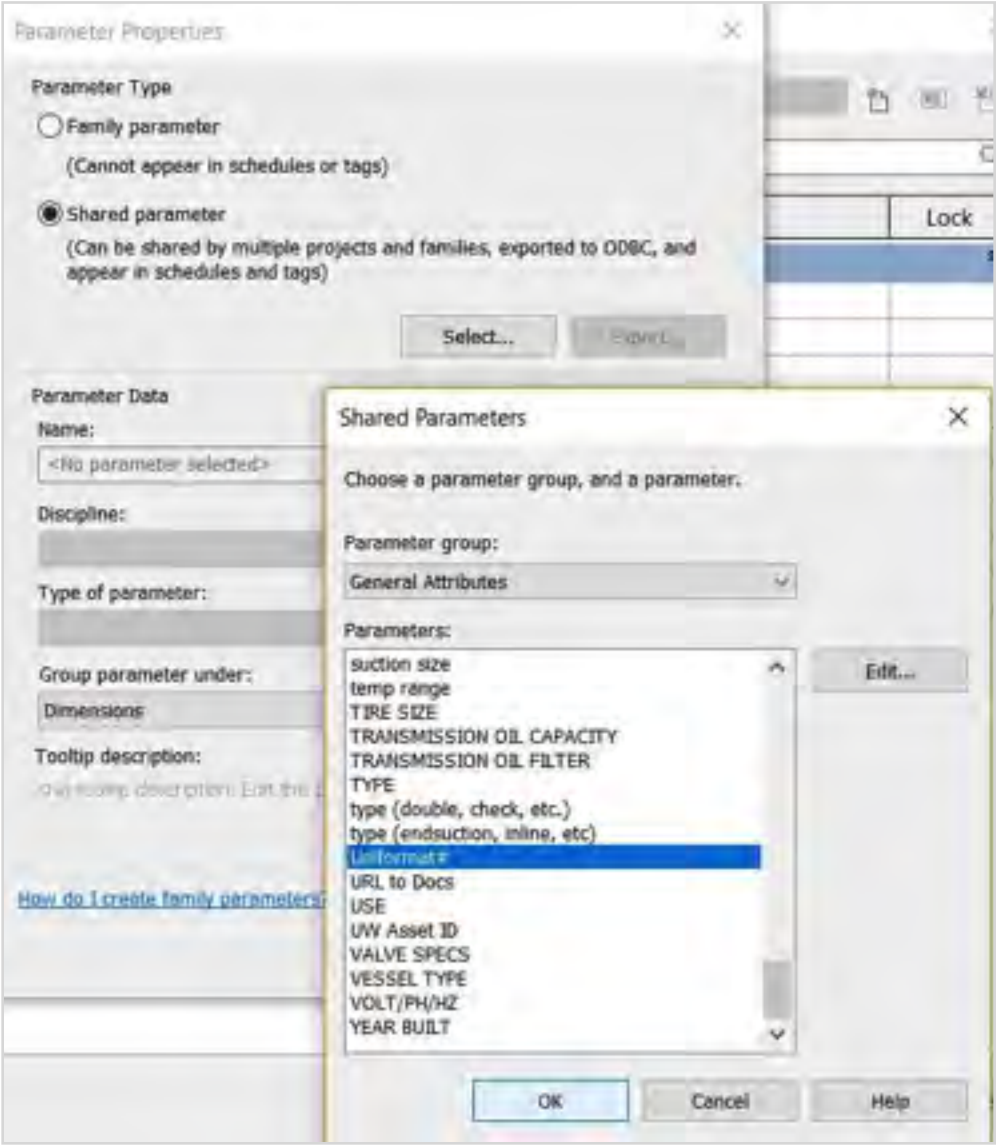



*Figure 54 - Select Shared Parameter and click Select to choose a parameter for this family.*

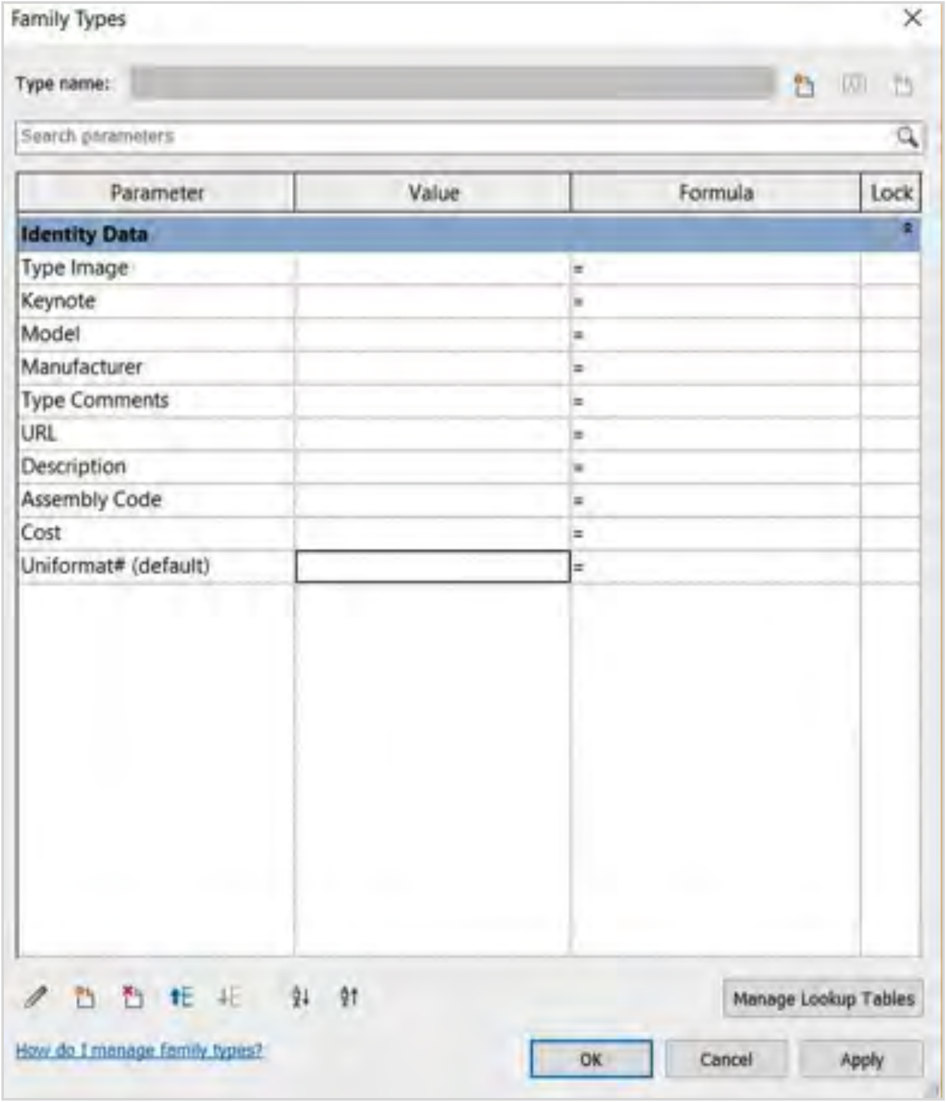



*Figure 55 – The parameter added is to the identity data attributes of the family.*

### 4.1.3. Enabling Room Calculation Point

To make all families aware of their location, the users must enable the Room Calculation Point.

- From the Project Browser under Floor Plans, open Ref. Level view.
- In the Properties window, scroll down and enable the Room Calculation Point. A green point will appear on the viewpoint when the Room Calculation Point is enabled (Figure 56).
- Save the generic asset file.

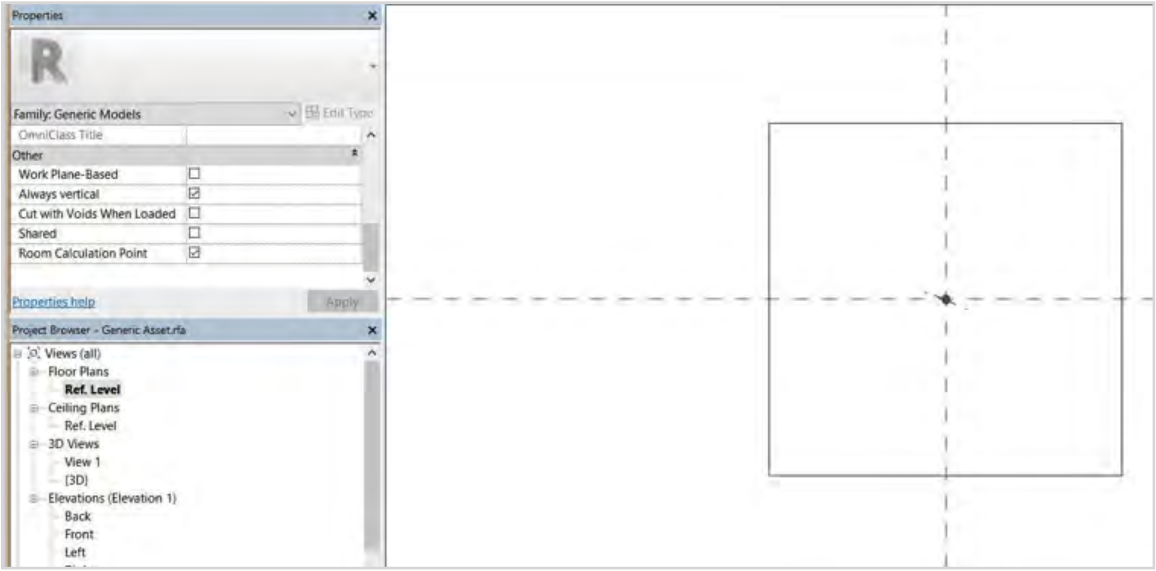



*Figure 56 – Enabling Room Calculation Point from the Properties window.*

### 4.1.4. Creating Generic Asset Systems

After saving and creating a back-up file for the generic asset, the users must use multiple copies of it to create generic asset system families. These families serve as the basis for creating different asset categories that are listed under that system. Therefore, our goal is to have a separate family file for each asset system (Table 3). For each system, the processes outlined below must be repeated.

| Asset system family templates | Colors to be assigned as Revit materials |
|---|---|
| ARCH, Building, EGRESS, ROOF, Furniture, Access Panel | White |
| Security, Safety, EHS | Yellow |
| Site | Dark Green |
| HVAC | Purple |
| Pump, Plumb, Irrigation, Utility | Blue |
| N/A | Revit Default Color |
| Fire Alarm, Fire, Fire Supply | Orange |
| Food Service | Light Blue |
| Electrical | Light Green |

*Table 3 – Asset systems and their suggested colors.*

• From the Create tab's Properties panel, open Family Types and fill in the asset system name in the Formula field with quotation marks. This makes this value non-editable in the rebaselining project environment. As the Formula column does not recognize text without quotation marks, quotation marks should be inserted around the text typed in Formula field (Figure 57).

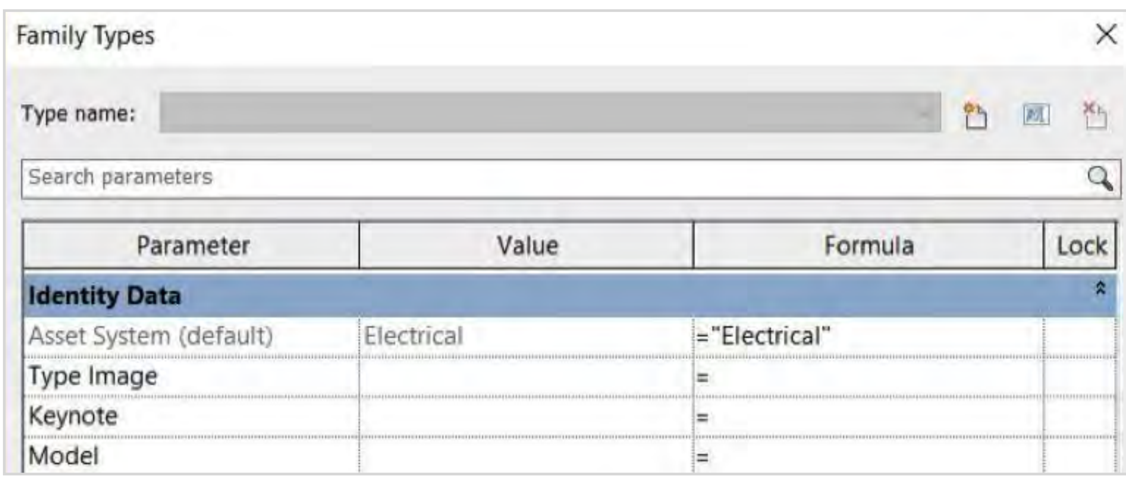



*Figure 57 – Filling in System-Level Attributes.*

In the next step, we want to assign a color to the box we created earlier. To do so, we need to create a Revit material.

- From the Manage tab, open Materials and choose Create New Material (Figure 58 and Figure 59).
- In the Identity tab of the Material Browser, rename the default name of the new material with the asset system name (Figure 60).
- Select the appropriate colors in the Graphics and Appearance tabs and click OK (Figure 61).
- Select the box in the generic asset family. From the Properties window, click on Material field and select the material you created (Figure 62).
- Open the Default 3D View and apply the Shaded visual style to see the color assigned to the geometry (Figure 62).
- Repeat the steps for all asset systems listed in Table 3.


*Figure 58 – Open Materials.*

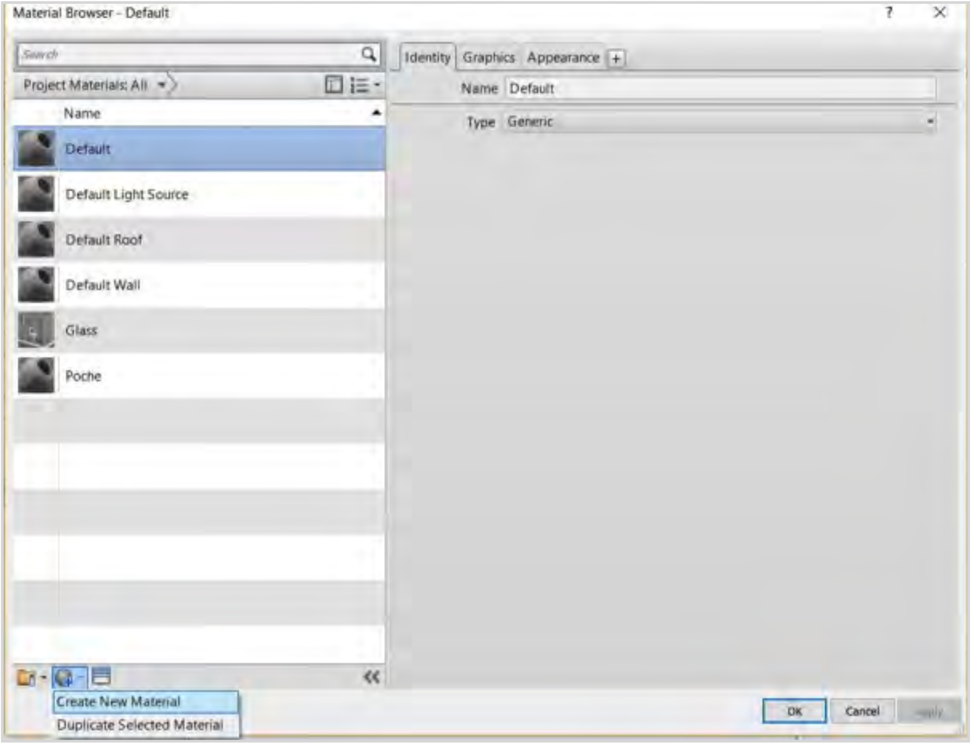



*Figure 59 – Create a New Material from the lower toolbar.*

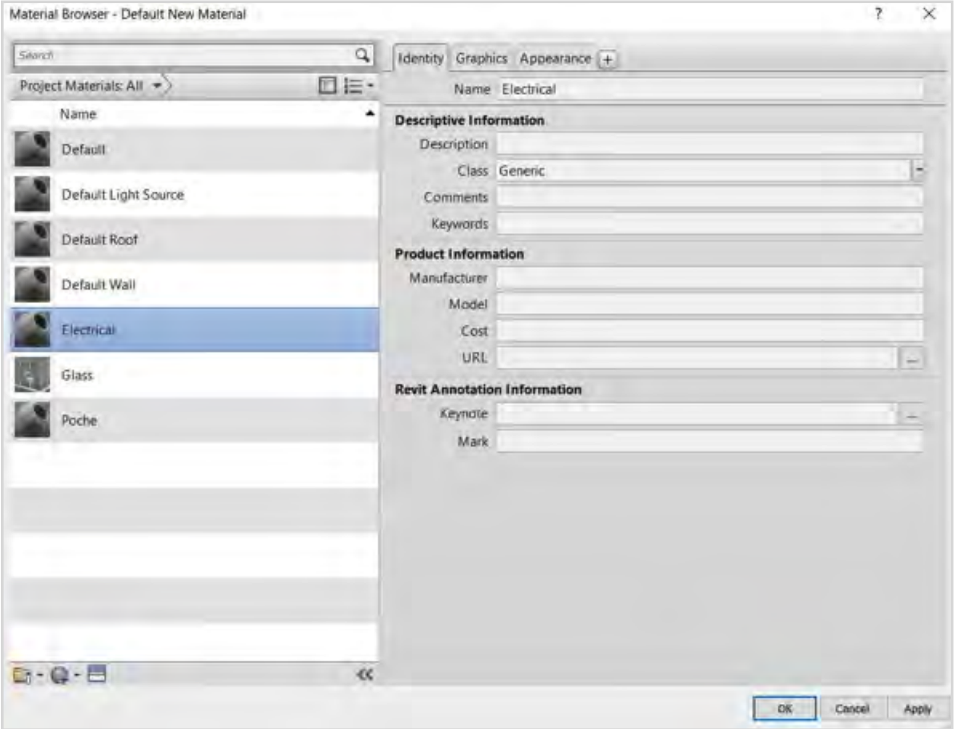



*Figure 60 – Rename the material in Identity tab.*

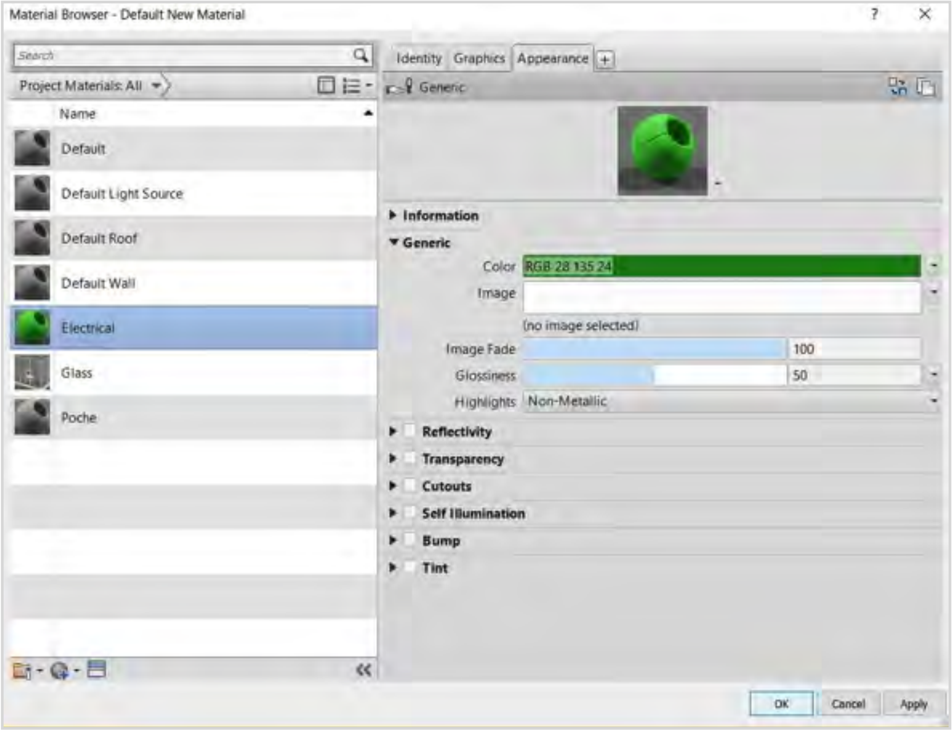



*Figure 61 – Assign a new color to the material in Graphics and Appearance tabs.*

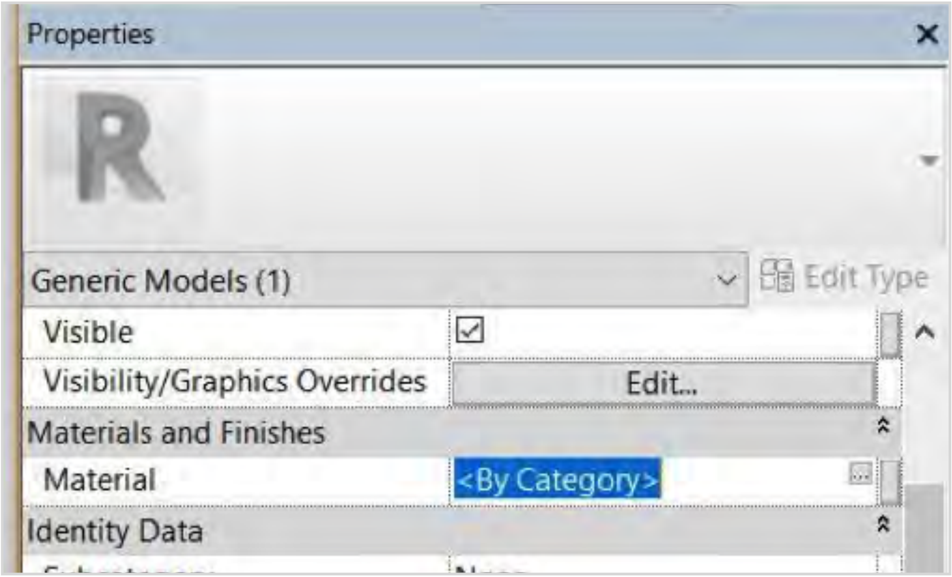



*Figure 62 – Assign the material to the box.*

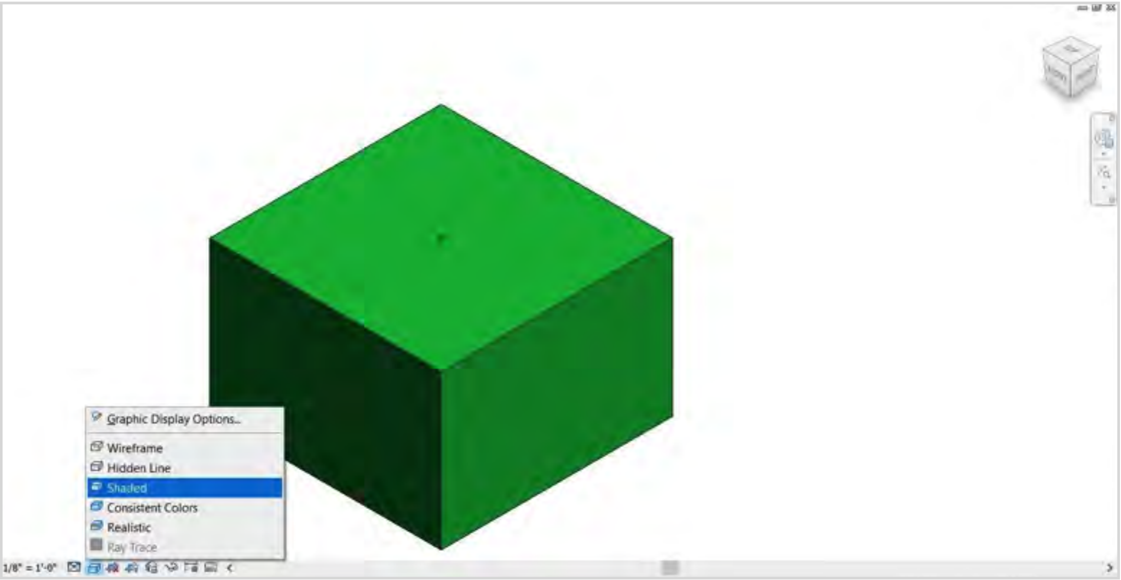



*Figure 63 – Choose Default 3D View and apply Shaded visual style to see the color of the geometry.*

### 4.1.5. Creating Asset Categories from Asset System Family Files

In this section, we need to create asset families based on the shared parameters file and generic asset system families we created in sections 3.1.1, 3.1.2 and 3.1.3. For each asset category listed in the asset data delivery guide, an asset family will be created. In the Family Editor environment, our goal is to add unique attributes of each asset to its family and add a symbolic 3D geometry that visually represents the asset. We use the asset data delivery guide as the basis for creating asset categories and asset attributes (Figure 64).

| | A | B | C | D | E |
|---|---|---|---|---|---|
| 1 | UW Asset Group "Type-Category" | Asset Group Description | Classification | Comments | Attributes |
| 2 | UW2000081 | ACCESS PANELS | Uniformat | | capacity |
| 3 | | | D7010.10 | | installed capacity |
| 4 | | | Master Format | | power supply |
| 5 | | | 28 13 00 | | remote power booster? |
| 6 | | | Omni-class | | remote battery backup? |
| 7 | | | 23-29 13 00 | | |
| 8 | | | | | |
| 9 | UW2000002 | CEILING TILE SYSTEM | Uniformat | ONE PER BUILDING | |
| 10 | | | C1070.10 | | |
| 11 | | | Master Format | | |
| 12 | | | 09 51 00 | | |
| 13 | | | Omni-class | | |
| 14 | | | 23-15 19 15 | | |
| 15 | | | | | |
| 16 | UW2000023 | FLOORS SYSTEM | Uniformat | ONE PER BUILDING | |
| 17 | | | C2030 | | |
| 18 | | | Master Format | | |
| 19 | | | 09 60 00 | | |
| 20 | | | Omni-class | | |
| 21 | | | 23-15 17 00 | | |
| 22 | | | | | |
| 23 | UW2000056 | PARTITIANS, MOTORIZED | Uniformat | | |
| 24 | | | C1010.50 | | |
| 25 | | | Master Format | | |
| 26 | | | 10 22 00 | | |
| 27 | | | Omni-class | | |
| 28 | | | 23-15 11 17 | | |
| 29 | | | | | |
| 30 | UW1000004 | DOOR, EXTERIOR | Uniformat | | |

Asset Type Matrix | Asset Group Name

*Figure 64 – Asset categories and asset attributes in the asset data delivery guide.*

• Open generic asset system families (e.g., architectural, electrical, HVAC, etc.).

• Save a new asset family for each asset category (e.g., access panel, ceiling tile, VAV Box).

• Open each asset family.

• From the Create tab's Properties panel, open Family Types.  Fill in the Formula field of asset classification attributes (e.g., Asset Description, Asset System, Asset ID, Masterformat#, OmniClass#, UNIFORMAT#)

• From the Shared Parameters file, add asset specific attributes or unique attributes of that asset category (see examples in Figure 66). Since values for these attributes may vary across projects, we do not need to enter any value for these attributes in Family Editor environment (Figure 67).

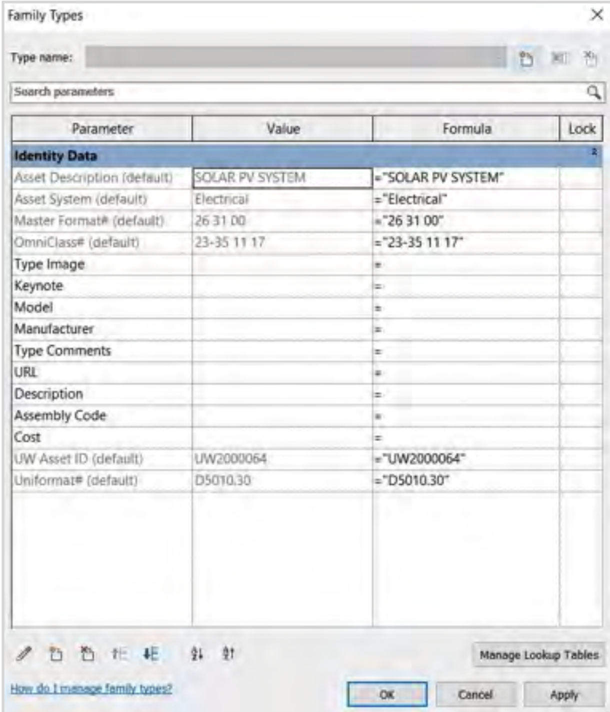



*Figure 65 – Fill in Formula field for asset classification attributes.*

| | A | B | C | D | E | F |
|---|---|---|---|---|---|---|
| 1 | UW Asset Group "Type-Category" | Asset Group Description | Classification | Comments | Attributes | System Sorting "System Name" |
| 367 | | | 23-17 21 13 17 | | | ELEC |
| 368 | | | | | | |
| 369 | UW2000064 | SOLAR PV SYSTEM | Uniformat | ONE PER BUILDING | power supply | ELEC |
| 370 | | | D5010.30 | | electrical panel name* | ELEC |
| 371 | | | Master Format | | capacity* | ELEC |
| 372 | | | 26 31 00 | | cell type | ELEC |
| 373 | | | Omni-class | | power supply | ELEC |
| 374 | | | 23-35 11 17 | | number of cells | ELEC |

*Figure 66 – Example for asset specific attributes of solar PV Systems.*

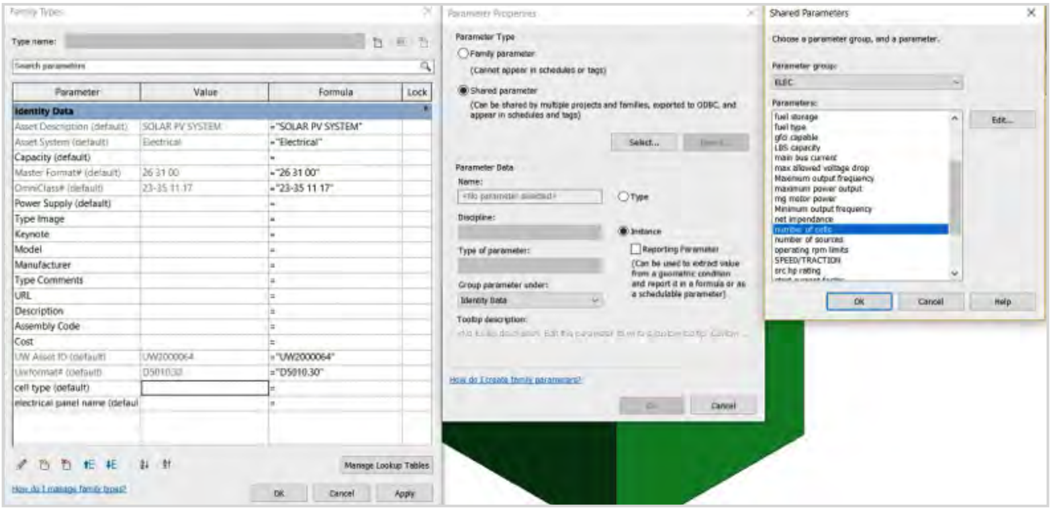



*Figure 67 – Adding asset specific attributes for solar PV system.*

In the next step, we attempt to add symbols or geometries to the asset families in order to facilitate the visual identification of them in rebaselining projects, where we may see many different asset instances in rebaselining information models. The users have to find a relevant geometry that meaningfully represents that asset in a plan view. For example, the geometry should look like a chiller, valve, or other listed assets in the plan view, although we may need to rotate the geometries to make them appropriate for the plan view (i.e., a fire extinguisher from a top view may not be very clear, so we may rotate it to have its side visible in the plan view of the family). These geometries should be simple, but informative. We do not need highly detailed models as they would have very large files and loading them into Revit takes a lot of memory.

• Find low LOD symbols/geometries in conventional CAD formats (e.g., DWG, SKP). We used 3Dwarehouse to download symbols/geometries in Sketchup format (SKP): https://3dwarehouse.sketchup.com (Figure 68). After downloading and opening the files, the users must save them as Sketchup 7 version to be used in Revit.

• In the Revit family editor environment, in the asset family file saved for each asset category, open Insert tab, click Import CAD (Figure 69).

• Set filetype first (e.g., to SKP) and adjust other settings as shown and import the file (Figure 70).

• Select the imported geometry and unpin it by clicking on the pin icon. Using different views available in the Project Browser (e.g., Floor Plan, Elevation, 3D view) and the Modify tools in the Modify tab, move and scale the object to place it on top of the box.  Save the asset family (Figure 71, Figure 72, and Figure 73).

• Repeat these steps for adding asset specific attributes and low LOD geometry for other asset categories (Figure 74).

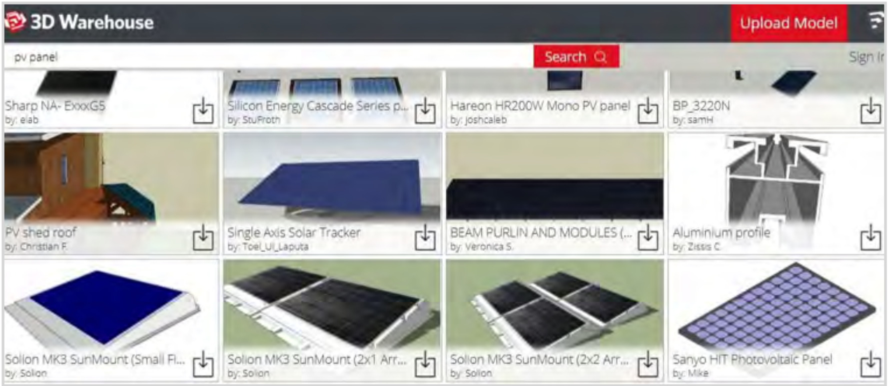



Figure 68 – Sketchup 3D warehouse website.

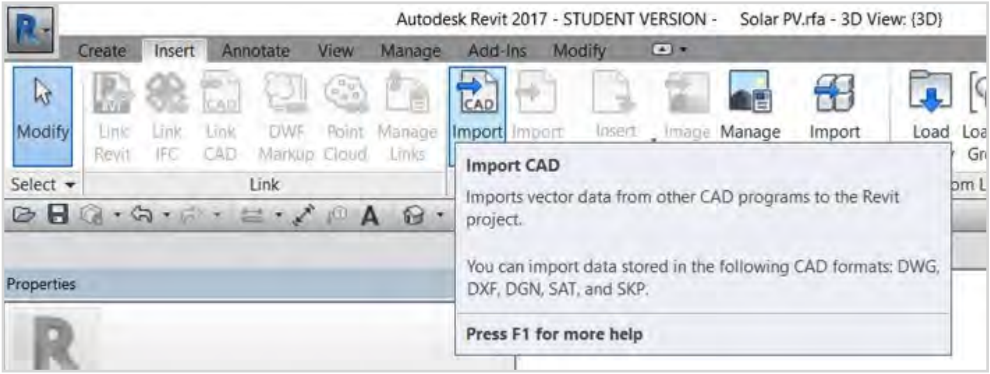



Figure 69 – Import CAD geometry into Revit family.

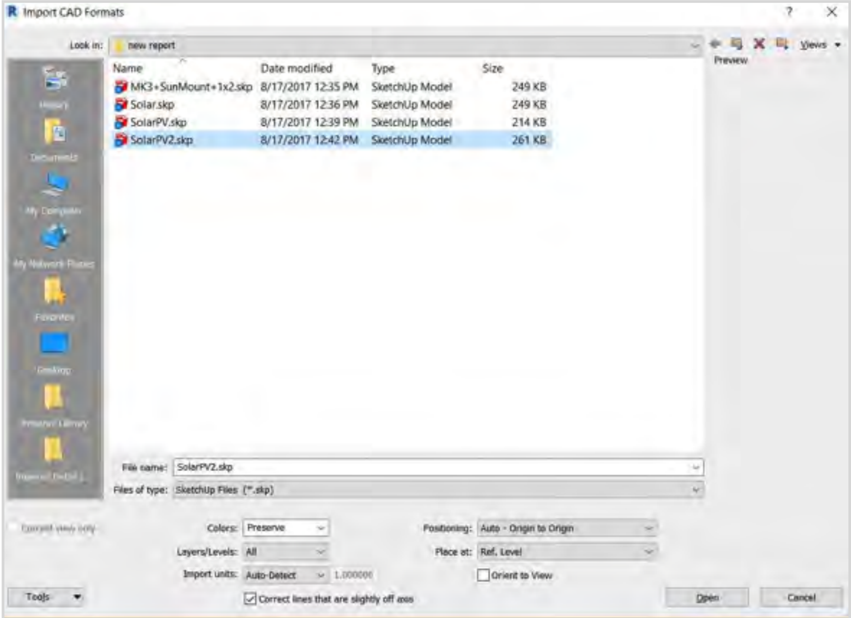



Figure 70 – Adjust the file type and open the Sketchup file.

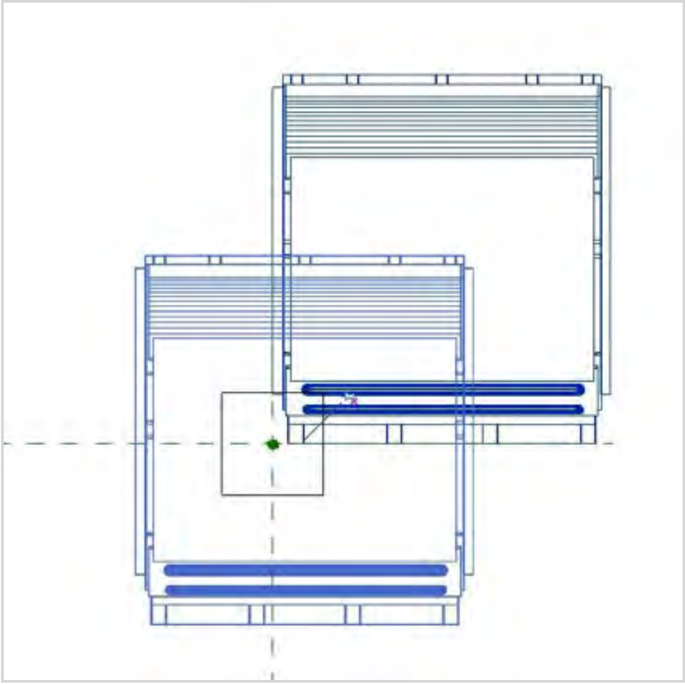


*Figure 71 – Unpin the imported geometry and move it.*

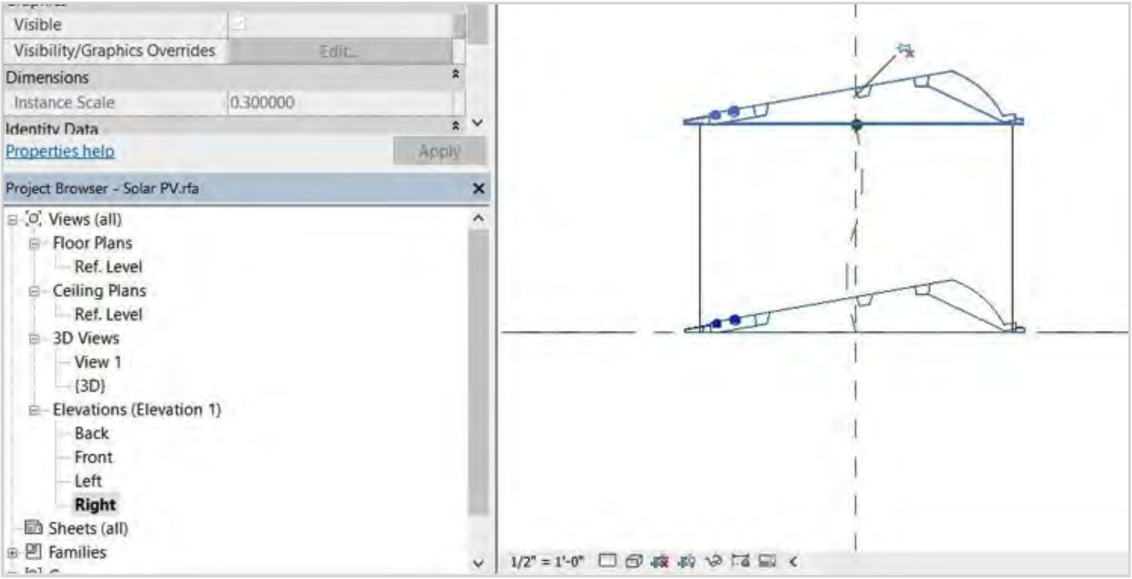



*Figure 72 – Use Right Elevation to position the imported geometry.*

*Figure 73 – Final appearance of the asset family.*

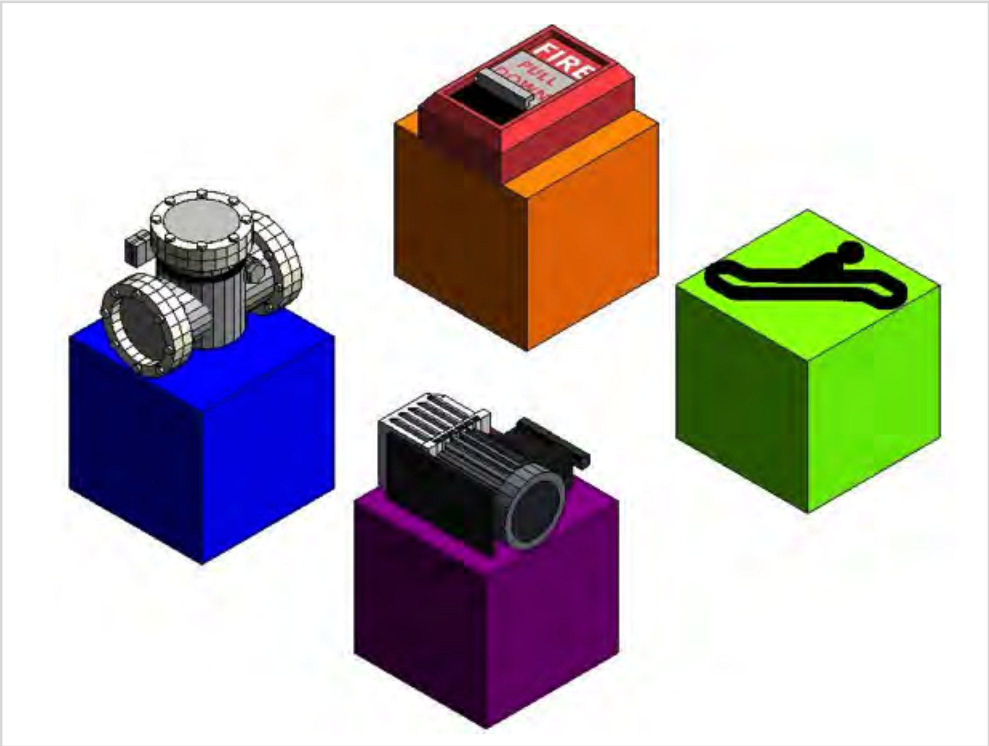



*Figure 74: Examples of different asset categories from different asset systems with different colors, shapes, and geomteries based on the preferences of rebaselining team members.*

In the next step, our goal is to create a rebaselining template file and load all asset families into the template file. In the template file, we must also add those common attributes that are applicable to all asset categories regardless of their classification, system, or category (e.g., URL to Documentations).

• Open a new Revit Project (Figure 75). Use the Architectural Template.

• From Insert tab, click Load Family (Figure 76). In the pop-up window, go to the folder in which all asset families are saved. Select all files and click Open (Figure 77). Note that loading families may take several minutes.

• From the Architecture tab, click Component. In the Properties window, click on the drop-down menu to ensure that all asset families are loaded into Revit (Figure 78).

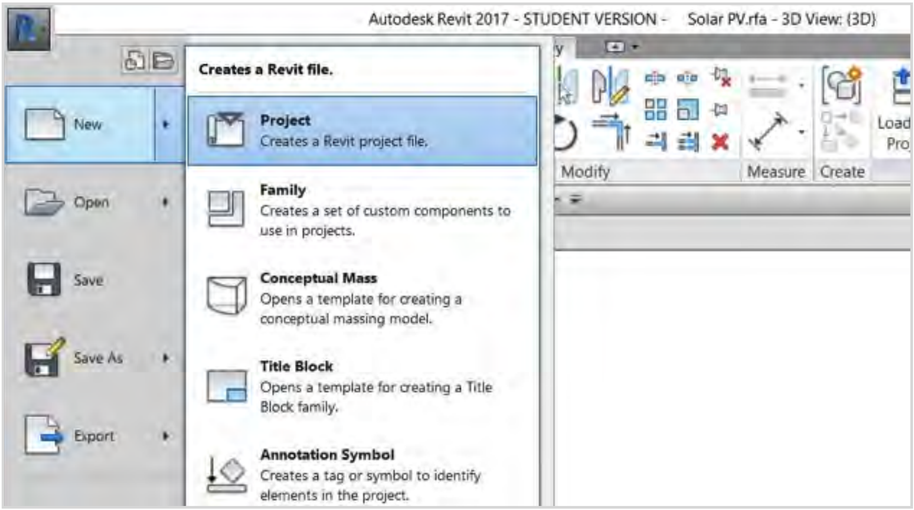



*Figure 75 – Opening a new Revit project.*

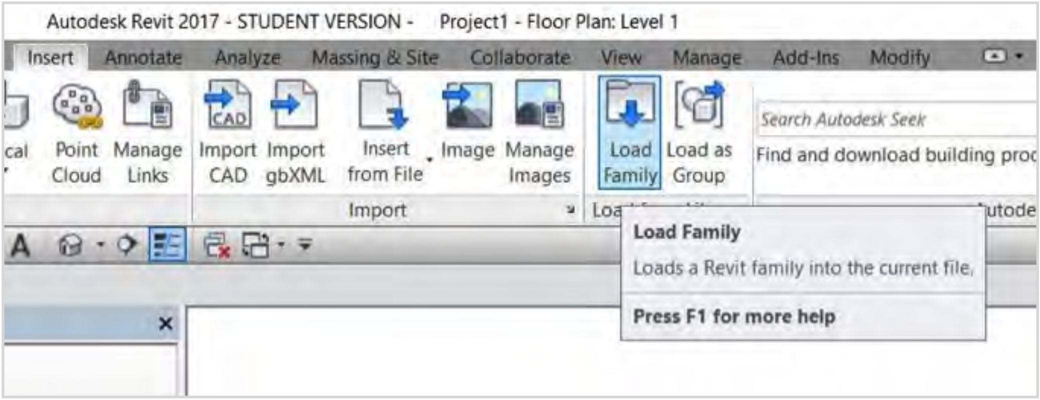



*Figure 76 – Click Load Family.*

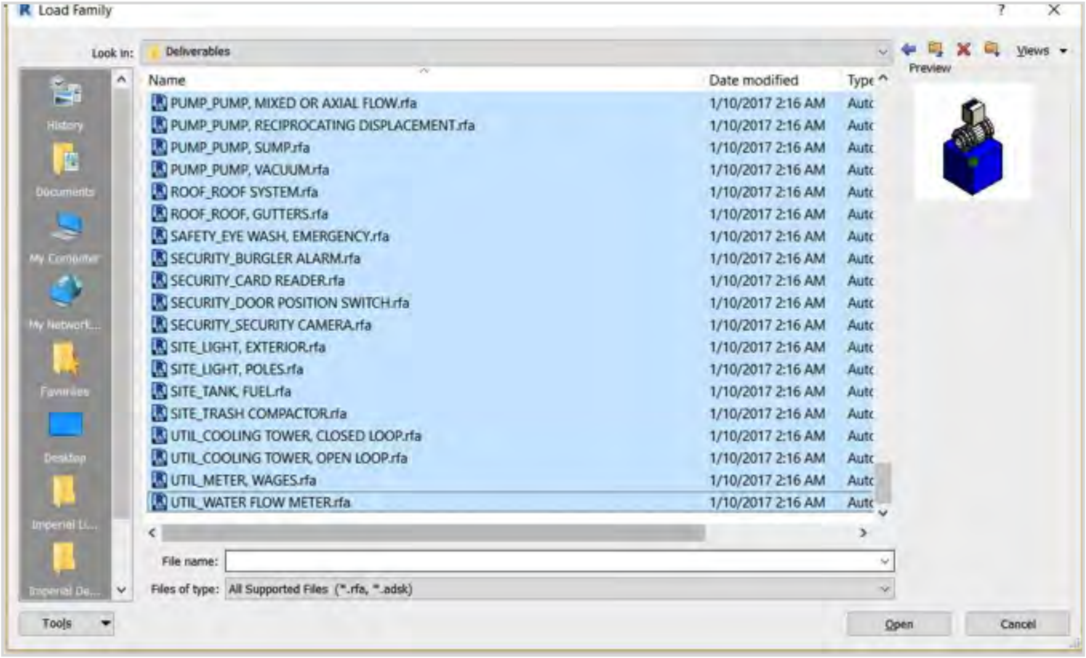



*Figure 77 – Select all asset families and click Open.*

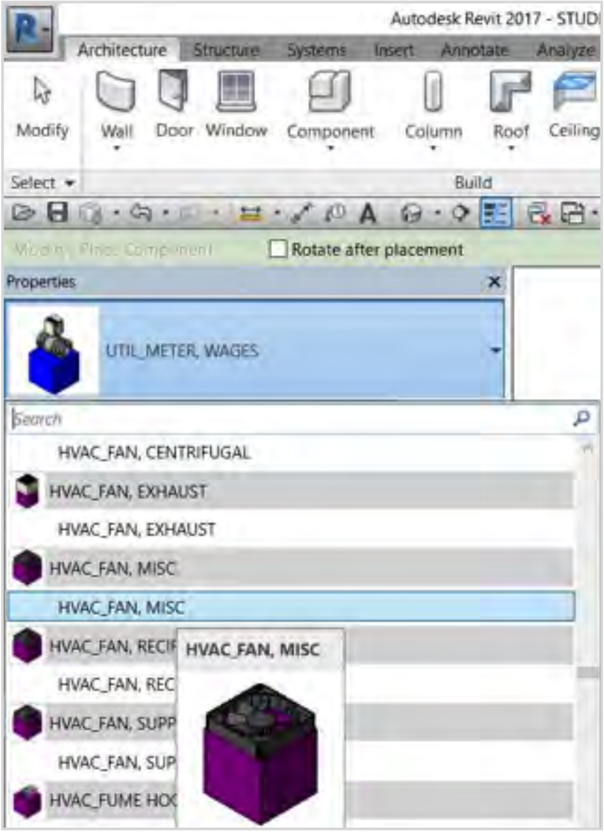



*Figure 78 – Accessing the asset families from the Properties window (after enabling the Architecture tab and Component command).*

• From the Manage tab, select the Settings panel and click Project Parameters. In the Project Parameters pop-up window, click Add (Figure 79).

• In the Parameter Properties pop-up window, choose Shared Parameters as the source of parameter we want to use and click Select to choose the required parameters (Figure 80). For simplicity, we want to use all parameters as Instance parameters and their values can vary by group instance. The parameters should be grouped under Identity Data. From the Categories section, enable Generic Models, Walls, and Doors because we want to see the parameters in all asset categories (we modeled them as Generic Models) in addition to walls and doors (which are used in this workflow for creating a 3D layout for facilities and creating rooms). Click OK to apply the changes.

• All attributes, including asset classification attributes, asset specific attributes, and common attributes are now available in the Properties window for each asset category and asset instance (Figure 81).

• From the application menu, save this Revit file either a project or as a template (Figure 82).

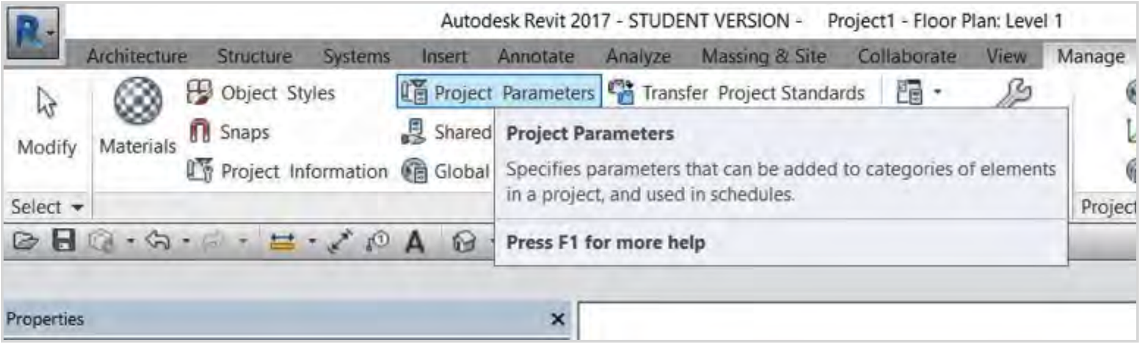



*Figure 79 – Open Project Parameters.*

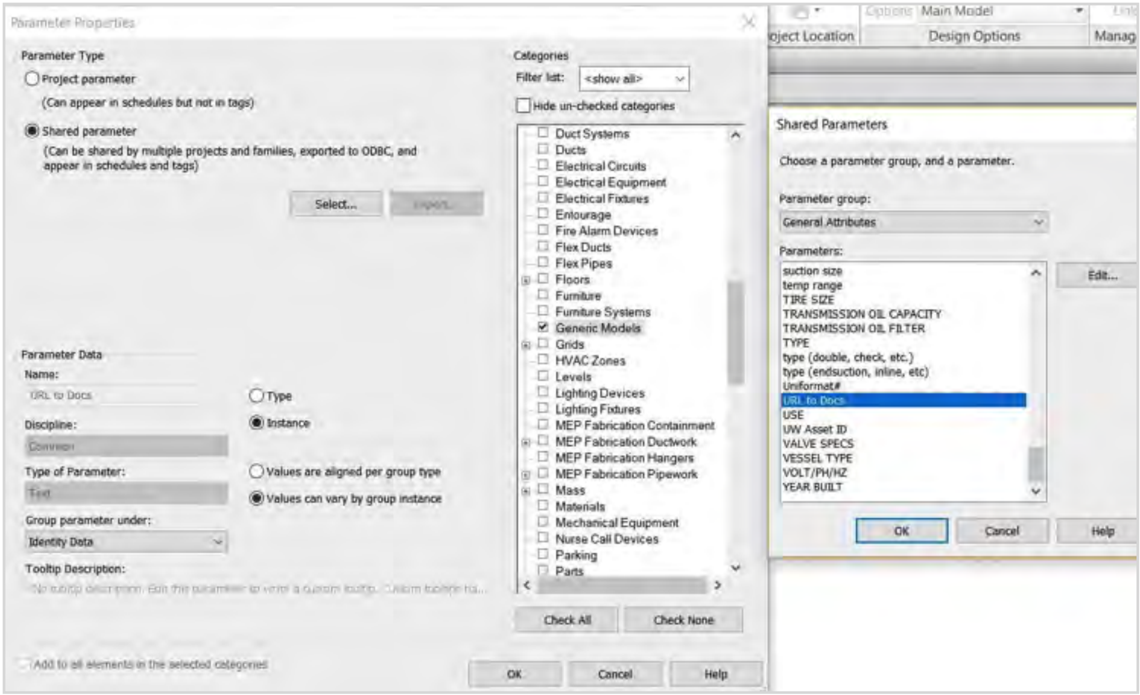



*Figure 80 – Adjust settings in Parameter Properties and add attributes from Shared Parameters.*

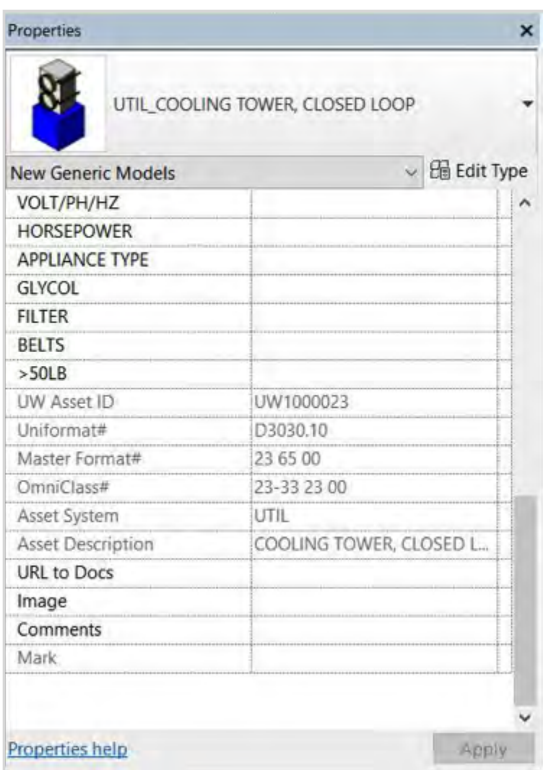



*Figure 81 – All attributes added to all asset categories in the project template.*


*Figure 82 – Save the rebaselining template.*

## 4.2. Phase 2: Collecting Asset Data from Existing Documentation Archives

In phase 2 (Figure 83), users must ensure that all project drawings/documentations are appended into the rebaselining software so that users can place asset instances as an overlay. Users should also extract asset data from different sources (e.g., manuals, drawings, schedules, brochures) and aggregate them in data fields that are linked to the assets. The rebaselining workflow is limited by data collection issues, such as lost or missing documents, unavailable data (not commissioned in project documents), and conflicting data (e.g., conflicts between O&M manuals and schedules on drawings). These issues must be documented in a rebaselining log file in order to be coordinated with asset management staff and, if possible, reconciled during field verification. Facilities management staff may have personal knowledge of assets or keep personal files in their daily practice that can also be used to refine the asset dataset during rebaselining. After this coordination process, asset data models can be accordingly updated.

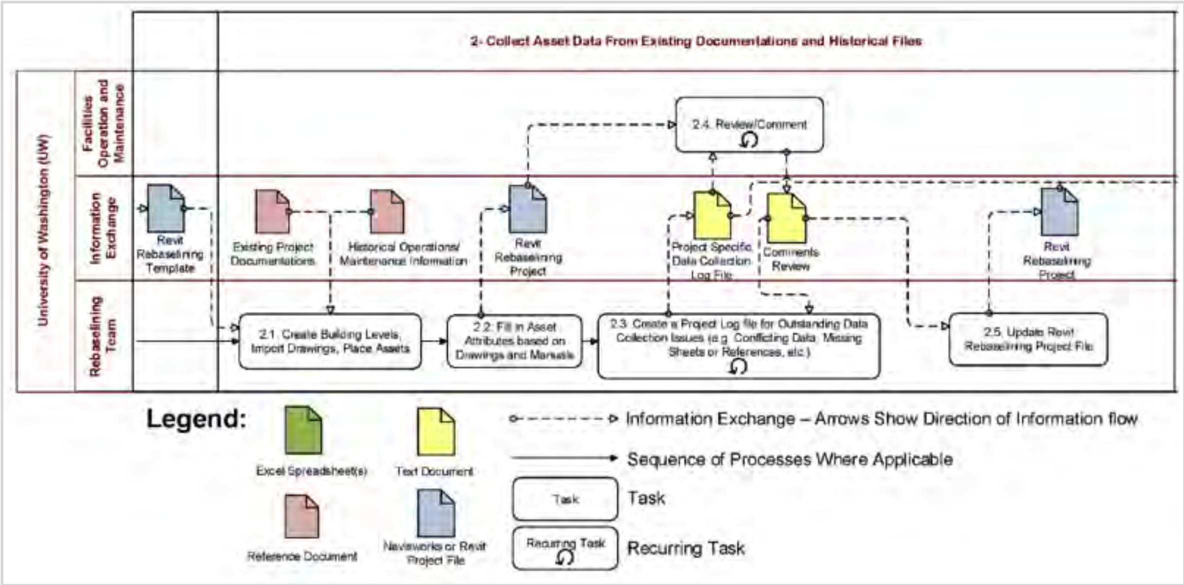



*Figure 83 – Phase 2 in the rebaselining workflow: Collecting asset data.*

## 4.2.1. Add Drawings to the Rebaselining Project

• Open the rebaselining template and save it as a new rebaselining project.

• From Project Browser, open one of the Elevation views and double click on the name of the view (Figure 84) and add building levels if necessary using the Level Command on the Architecture tab's Datum panel (Figure 85).

• From Project Browser, open one floor plan at a time and import relevant CAD drawings onto the opened floor plan. From Insert tab, click Import CAD to import DWG files (set filetype, unit of measurement, and placement options in the pop-up window). This process can be repeated if there are multiple drawings for each building level (e.g., Architectural, Mechanical, Electrical).


*Figure 84 – Open one Elevation view in the Project Browser.*

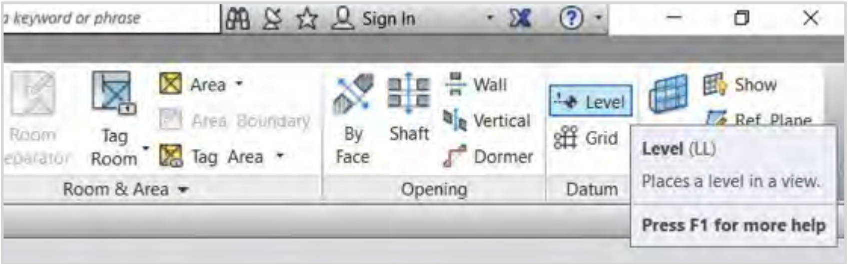



*Figure 85 – Use Level command from the Architecture tab to add to building levels.*

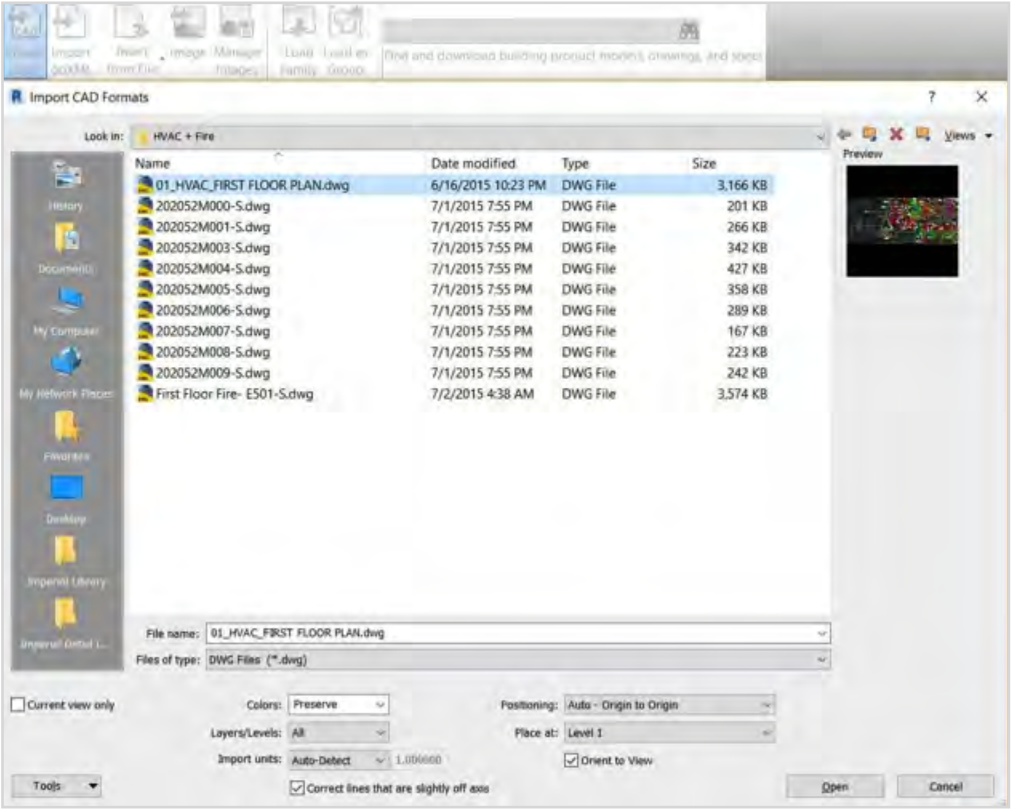



*Figure 86 – Importing DWG drawing onto a floor plan.*

• After importing drawings, click the Component command on the Architecture tab. From Properties window, select relevant asset categories and place asset instances on relevant symbols on the drawings (Figure 87). Repeat the process for all required asset categories and all asset instances identifiable in project documentations (Figure 88).

• Select one or more asset instance and fill in attribute fields in Properties window (Figure 89). Repeat the process for all asset instances placed in the rebaselining data model (Figure 88).

• Users can create actual walls and doors to create a 3D layout of the building and add room information in Revit. As the asset instances are room aware (with a room calculation point), having rooms in Revit can facilitate extracting the location of asset instances.

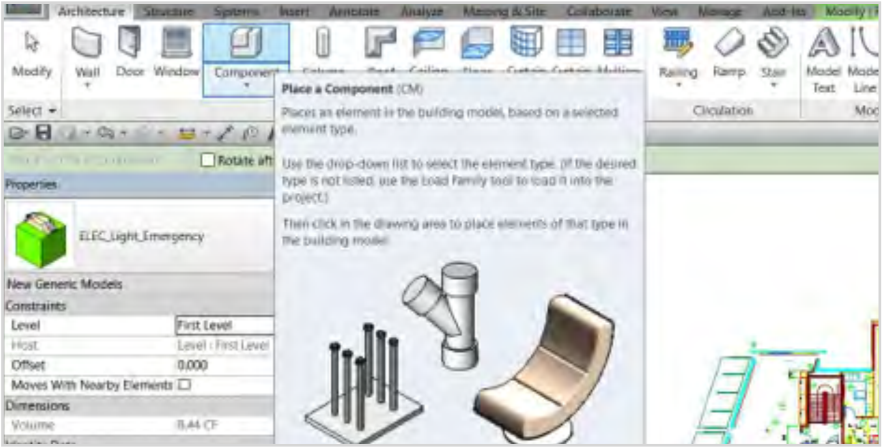



*Figure 87 - Placing asset instances on the drawing.*

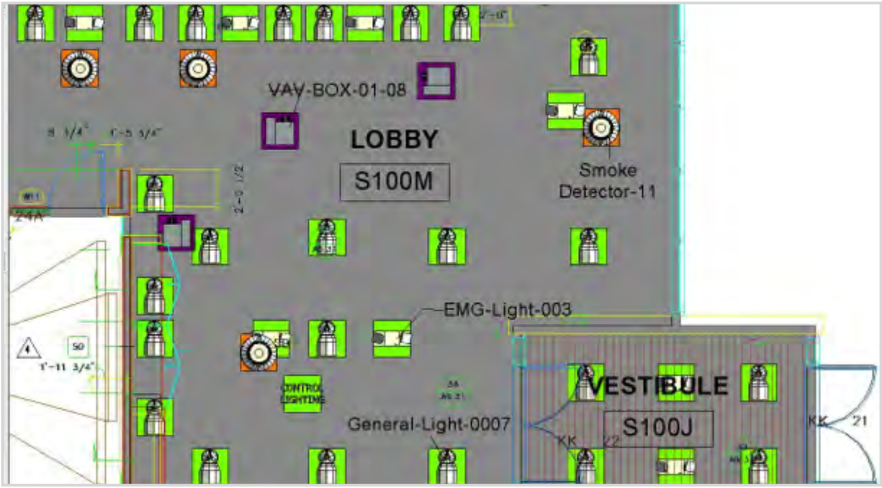



*Figure 88 – Example for placing multiple asset instances on a drawing.*

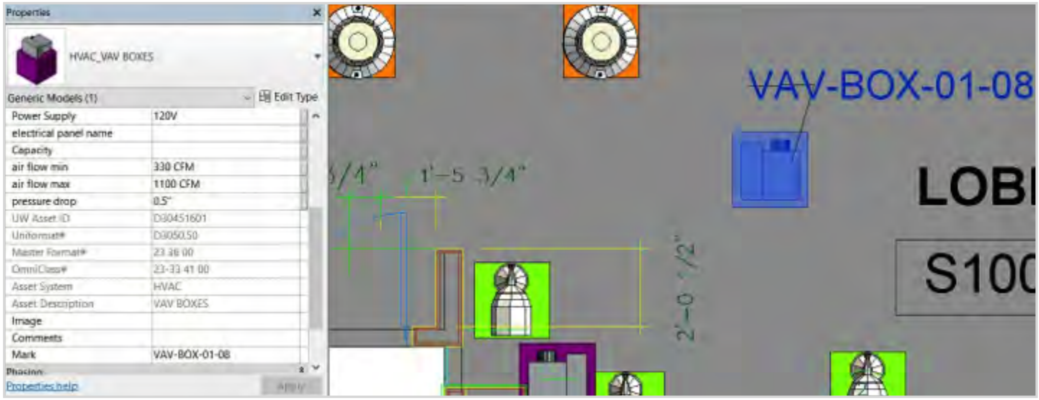



*Figure 89 – Filling in attribute fields of each asset instance in the Properties window.*

## 4.3. Phase 3: On-site Data Verification

Phase 3 in the rebaselining workflow (Figure 90) addresses field verification of asset data and updating asset data models or physical asset tags/plates if necessary. These processes are essential in the rebaselining workflow—and generally in asset management—because existing facilities often have undocumented asset modifications that cause interruptions in asset operation, maintenance, and renewal. Not only can this rebaselining process facilitate identifying and documenting these modifications in asset data models, but it is also an opportunity to compare documented data against physical asset tags to find data that are missing from manuals or drawings. For this reason, implementing this phase can significantly improve accuracy and reliability of asset data models.

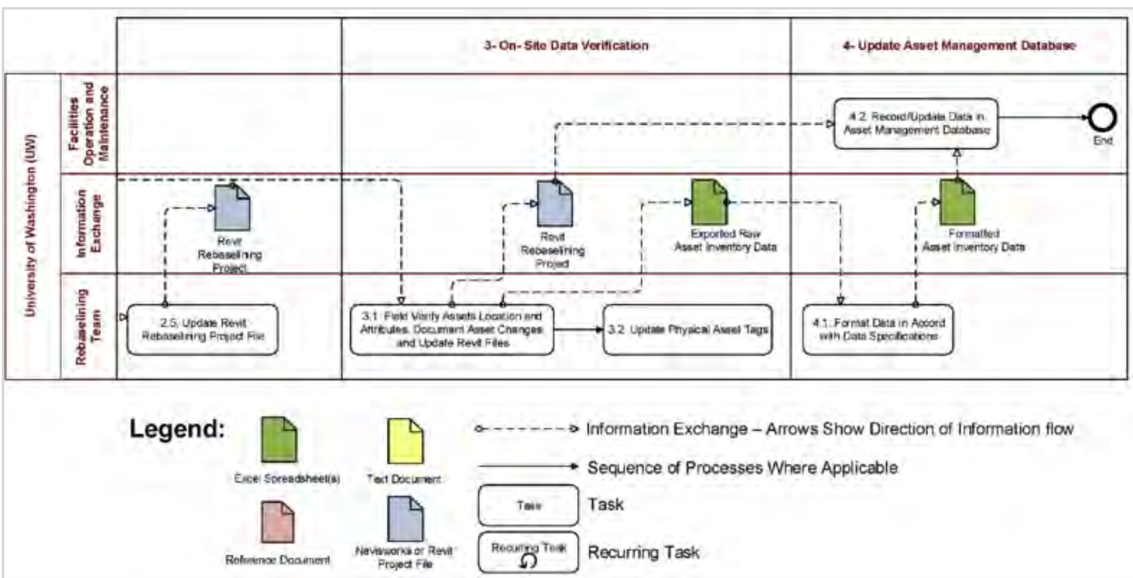



*Figure 90 – Phase 3 and Phase 4 in the rebaselining workflow: On-site data verification and updating the asset management database.*

## 4.4. Phase 4: Update Asset Management Database

Revit can export data from the verified asset data to raw data files in order to be prepared in the format required for CMMSs. As asset management staff may require a specific format for asset inventory data (e.g., COBie standard spreadsheet), phase 4 is included for transforming raw data files to the desired formats. The implementers must decide if the formatted asset inventory data must be machine-readable (for CMMS), human readable, or both because certain CMMSs may require specific data structures to support interoperable data transfer.

### 4.4.1. Exporting Asset Data to Raw Data File

After adding and verifying asset data, asset data can be exported to raw data files using the following process.

- From the View tab, select the Create panel. Select the Schedules menu and click Schedules/Quantities (Figure 91).
- In the pop-up window, select New Schedule and select Generic Models under the Category section. Click OK (Figure 92).
- In the Schedule Properties window, select all attributes from available fields, and click add parameters (green arrow) to include them in the schedule. click OK (Figure 93 and Figure 94).
- From the application menu, select the Export section and choose Reports > Schedule (Figure 95). The data file will be in TXT format and can be opened and processed in MS Excel.

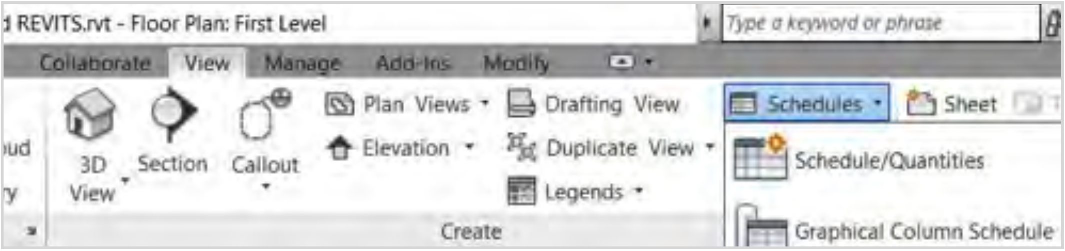



*Figure 91 – Create a new Schedule.*

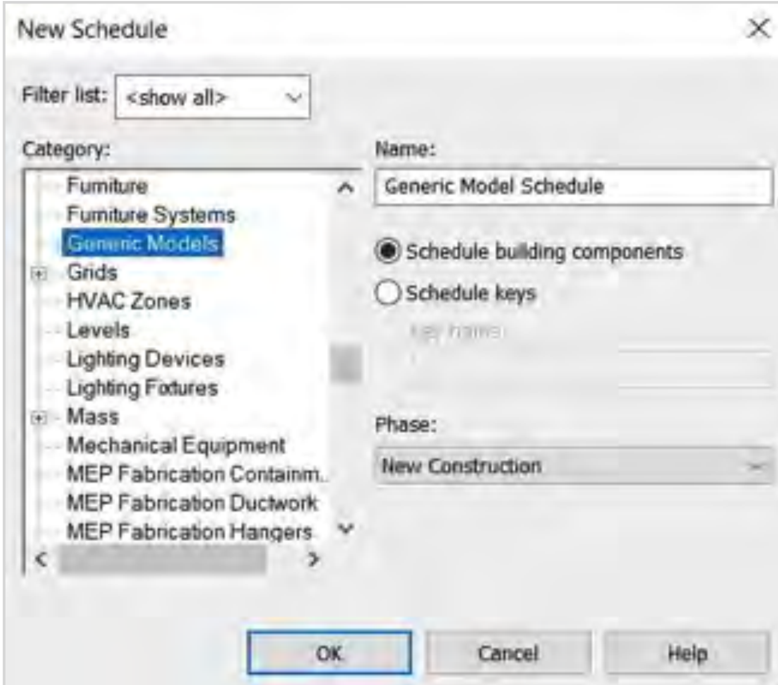



*Figure 92 – Data of Generic Models should be extracted.*

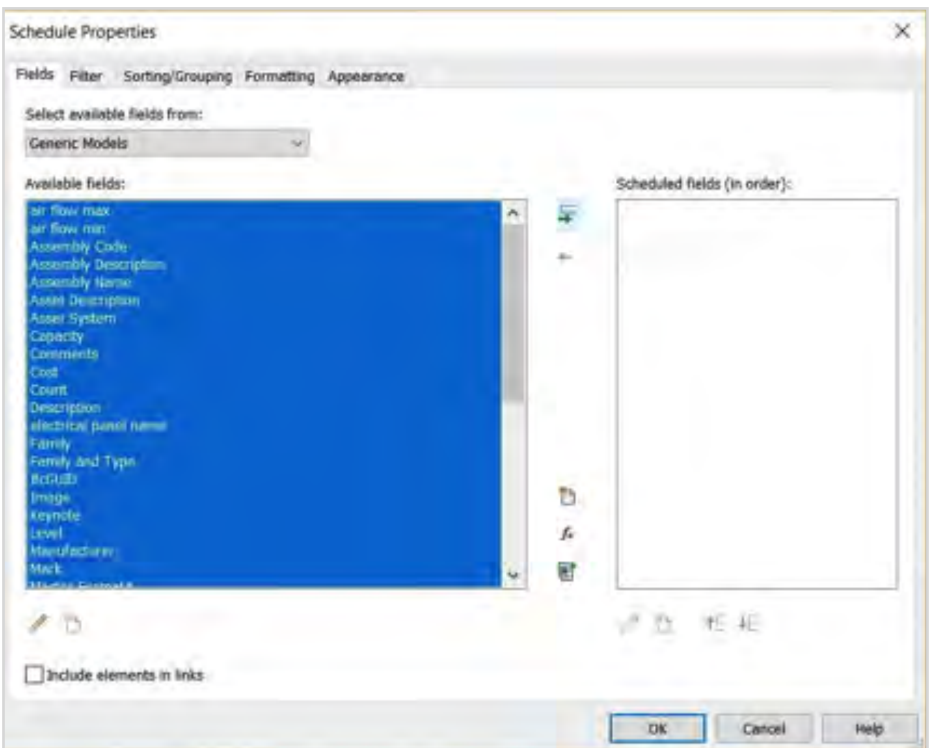



*Figure 93 – Select all available fields and click add parameters arrow.*

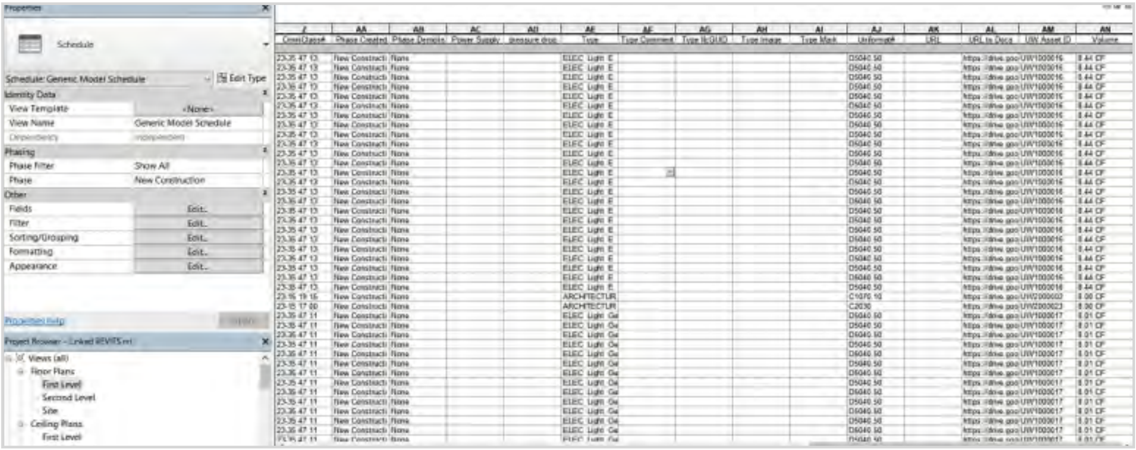


*Figure 94 – Schedule for asset data.*

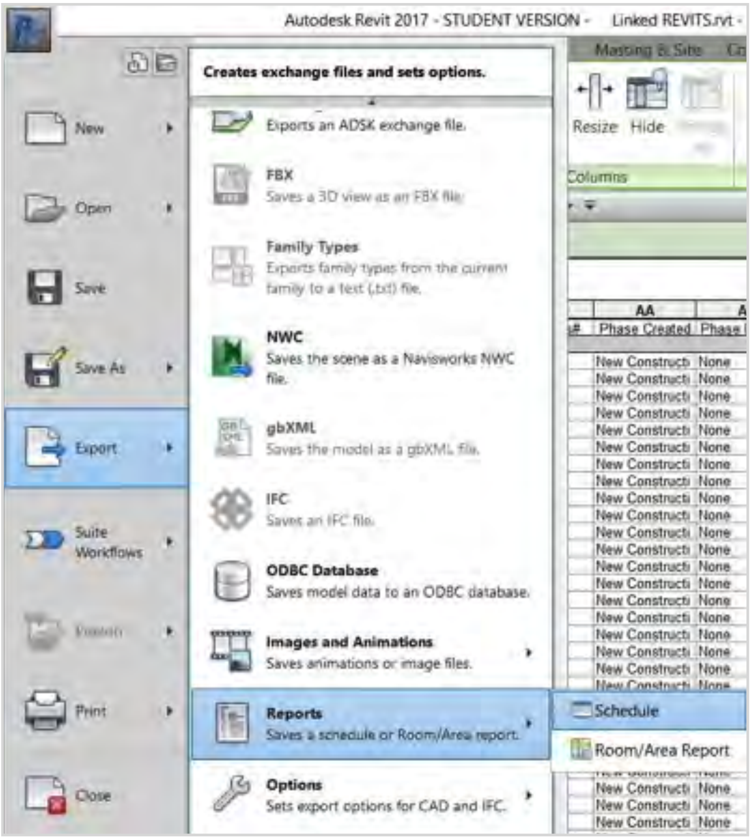



*Figure 95 – Export a schedule as a data file.*

# 5 CHALLENGES AND LIMITATIONS

Difficulty in accessing or finding existing project documentation (e.g., drawings, manuals) is an important challenge when collecting asset data as lost, partial, or damaged documents are not unusual in managing existing facilities. This challenge is salient even for newer facilities when a large number of digital documents do not have clear naming conventions or file organization and are not readily searchable. At a large scale, these issues can drastically influence the efficiency of a rebaselining workflow because standards for preparing project documentations, drafting and organizing drawings, and creating a projects manual (e.g., standards for symbols, attributes in schedules) may significantly vary across different projects and facilities.

Field verification of asset data can be challenging as well, especially if assets or their parts are not readily visible (e.g., buried pipes). In such cases, the staff cannot verify the exact location and attributes of assets, requiring them to rely on existing documentations until forensic analyses facilitate field verification of data. Furthermore, even accessible assets might not be field verifiable if they do not have physical tags. This issue becomes especially challenging when untagged assets in a space are physically similar, but might have different model numbers and/or manufacturers (e.g., valves in a mechanical room). Due to a lack of identification tags, team members might not be able to determine whether data available in the data sources are reliable or whether any asset has been replaced by a physically similar asset with different attributes.

**CERC** is where...

**THEORY** meets **PRACTICE**
**LAB** meets **FIELD**
**POSSIBILITIES** become **REALITIES**
and **CHANGE** happens

http://cm.be.uw.edu/cerc/