Connect the Dots: Architectural – Designing the Body

Marcus Kim, AIGA, Autodesk Consulting

AB4719-P This course covers the Architectural Workflows as a team begins developing a project model. In addition to typical starting procedures like template setup, model building and workset creation, the collaboration process and strategy will be covered from the Architectural point of view.

Learning Objectives

At the end of this class, you will be able to:

- Understand the workflow involved with large projects
- Understand the importance of effective setup and templates
- Understand an effective workset strategy
- Develop best practices for coordination between design teams, and other disciplines

About the Speaker

Mr. Kim has extensive experience working on, and managing the production of monumental scale architecture projects in both AutoCAD® and in Revit®. After working in design and technical architecture with Epstein-ISI in Chicago, Illinois, he found the opportunity to move into Design Technology management, where he quickly excelled, and was later recruited by SOM to manage the BIM Program in their Chicago office. At Autodesk, Mr. Kim is a Business Consultant, and has assisted clients from basic to advanced Revit concepts, Revit implementation, and technical training.

Mr. Kim has a BA in Architecture from the University of Illinois Chicago and an AAS in Computer Graphics from the American Academy of Art. Mr. Kim is also the chair of the first US Chapter of the internationally acclaimed organization, CGSociety.

Outline

0.0 Introduction

Since Revit's conception, it has been the goal of AEC firms to utilize BIM technology in ever growing projects. It wasn't too long ago that Revit projects were small in size, the hardware pushing the Revit files inadequate for the processing of the information. With every release of Revit, and advancements in hardware, the realization of large Revit projects has become a reality.

Success for these large projects are still difficult because the procedures and organizational process of how to develop a large project has yet to be formalized, i.e. companies are experimenting with large Revit project workflow procedures. It is the goal of this class, to convey key criteria that are necessary for the success of a large scale Revit project.

This section has been developed for the Architectural component of Revit. Though the lecture component of this class is 90 minutes and can cover only so much, the information contained in this packet goes beyond what was discussed and is relevant for the other flavors of Revit. Feel free to utilize the information in any manner you see fit to enhance the development of your Revit projects.

1.0 Understand the Workflow

In order to attempt a project in Revit, there must be an understanding of Revit's workflow process. To begin with, to even attempt a Revit project, considerations must be made for the necessary hardware. It is extremely unforgiving later in a project, if it is discovered that current company hardware standards cannot meet the performance demands of the Revit project file.

Similarly, even with the best hardware, when working on a Revit project, the design criteria for the model at each project phase may impact the workflow, not just in performance but in the expedient development of the Revit model. In many cases, companies who attempt the pilot Revit project, often times fail to realize such things as the Revit model's level of detail. For example, a schematic design Revit model which would only be documented at most an 1/8" = 1'-0" (or 1:100 for metric projects) is being developed at almost a construction document quality level. Revit projects being developed in this manner can lead to unnecessary expenditure of time as well as impact performance for a project during an early project phase where a high level of detail considerations are not necessary.

It is with these two premises that this section covers hardware requirements for a Revit project, as well as a point by point summary of what a Revit model needs to accomplish throughout the various phase of a project.

1.1 Hardware Requirements

When working on large Revit projects, it is not enough to have a workstation with the minimum system requirements. Revit projects, especially large projects push extreme amounts of data. With the advent of 64-bit hardware and its availability, it is becoming less of a critical issue. Similarly what is listed as minimum system requirements especially the amount of RAM was considered just five years ago the maximum RAM you could have in a 32-bit system.

The current minimum requirements for 32 bit systems

Microsoft® Windows® 7 32-bit Enterprise, Ultimate, Professional, or Home Premium; Microsoft® Windows Vista® 32-bit (SP2 or later) Enterprise, Ultimate, Business, or Home Premium; or Microsoft® Windows® XP (SP2 or later) Professional or Home*

Intel® Pentium® 4 or AMD Athlon™ dual core, 3.0 GHz (or higher) with SSE2 technology for Microsoft Windows 7 32-bit or Microsoft Windows Vista 32-bit (SP2 or later). Intel Pentium 4 or AMD Athlon dual core, 1.6 GHz (or higher) with SSE2 technology for Microsoft Windows XP (SP2 or later)

3 GB RAM

5 GB free disk space

The current minimum requirements for 64 bit systems

Microsoft® Windows® 7 64-bit Enterprise, Ultimate, Professional, or Home Premium; Microsoft® Windows Vista® 64-bit (SP2 or later) Enterprise, Ultimate, Business, or Home Premium; or Microsoft® Windows® XP Professional x64 edition (SP2 or later).*

Intel® Pentium® 4 or AMD Athlon™ dual core, 3.0 GHz (or higher) with SSE2 technology for Microsoft Windows 7 64-bit or Microsoft Windows Vista 64-bit (SP2 or later). Intel Pentium 4 or AMD Athlon dual core, 1.6 GHz (or higher) with SSE2 technology for Microsoft Windows XP Professional x64 edition (SP2 or later)

3 GB RAM

5 GB free disk space

The following system recommendation is best for large Revit projects.

Since the adoption of 64-bit systems, the speed of the processors and the available amount of RAM has drastically increased. At a certain point the workstations that are provided to Revit users can be over-kill, so a fine balance should be considered that weighs cost vs performance.

Microsoft® Windows® 7 64-bit Enterprise, Ultimate, Professional, or Home Premium, Microsoft® Windows® Vista 64-bit (SP2 or later) Enterprise, Ultimate, Business, or Home Premium edition, or Microsoft® Windows® XP Professional x64 edition (SP2 or later)

Quad Core Intel® Xeon® Processor (2.50 GHz, 2X6M L2, 1333) or equivalent AMD processor

8 GB RAM (or more)

1,280 x 1,024 monitor with true color

1 GB (or more) DirectX® 10-capable graphics card with Shader Model 3 for advanced graphics.

Tips and Recommendations

When considering a workstation for your future projects, remember that the performance of the workstation and the ability of it to be stable when processing vast amounts of information are paramount to success. This is not to say throw money at the hardware requirements but take careful consideration when selecting it. Some points to consider when purchasing a Revit workstation:

- **Processor Cores:**   Revit only utilizes multiple cores when tasked to render. All other Revit processes currently tasks only one core. Although the total amount of processing power for multiple cores may exceed any of your existing processors, due to the Revit's core usage this may be a downgrade for Revit. Also, when determining Processors, consider the other applications used, such as analysis applications, and visualization applications. What may be just good for Revit would not be as optimal for others

- **System Memory:** Best to fill all channels with same type of memory as opposed to mixing types or leaving channels empty. Fill all memory slots (dual / triple channel) as much "dual-rank" (2-, 4, or 8-GB) memory as you can afford. Memory availability is top priority for large Revit projects. In my opinion the standard workstation has 16GB of memory. Have enough memory slots to allow for future memory upgrades economically.

- **Graphics:** Use an Autodesk certified graphics card that meets rendering and output needs. A graphics card impacts performance in Revit when navigating. It will allow for the best graphical user experience. A Revit Autodesk certified graphics card is recommended. Both AMD and NVIDIA has comparable cards, personally I do not have a preference as long as they are certified. With that said, the Architecture users may have a need for higher end graphics cards due to their need for visualization tasks. Consider the possibility of having two machine configurations. One for Architects, one for engineers

- **Storage:** 7200-RPM SATA disks unless large-volume rendering is required Also this will impact performance of certain Revit tasks beyond just rendering. Such as the regeneration of multiple views, printing of large multi-model Revit projects may force Revit to tap into the Virtual RAM.

- **Operating Systems:** Should be using a 64-Bit Operating system to deliver maximum memory performance. Remember that the total allowable RAM usage is dependent on the version of the operating for example in the case of Windows 7 the maximum RAM usage is as follows:

    - Starter: 8GB
    - Home Basic: 8GB
    - Home Premium: 16GB
    - Professional, Enterprise, Ultimate: 192GB

Closing Comments

Feedback taken from multiple customers has ranked following importance of hardware subsystems:

1. Memory
2. CPU a "very" close second
3. Graphics
4. Storage.

As a general statement, the super high end specifications of workstation hardware can be leveraged with best practices such as good workset management, selective workset loading/unloading, and proper modeling procedures for families. However, even with this statement, there will be instances where the top of the line workstation is necessary to produce the final deliverable. To elaborate, you may be able to have a spec for typical Revit user workstation that will accommodate all productions tasks, and have a spec for workstation that can be accessed via RDP, for the final deliverable tasks.

1.2 The Revit Project Workflow Defined

For multi-disciplined projects with a focus on Integrated Project Delivery (IPD), the Architecture workflow is typically the lead process that dictates a Revit project and coordinates all disciplines into one coordinated effort.

It is crucial from the onset of a Revit project, that the clarity of the design for the Revit project meets requirements for the final deliverable documentations. Due to the nature of the Architectural Design process, projects are typically separated into the following phases:

- Pre-Design (PD)
- Schematic Design (SD)
- Design Development (DD)
- Construction Document (CD)

The requirements of the Revit model are different for each phase, and therefore it is important to understand what is critical for development at each phase. With this understanding, it is possible to leverage Revit to meet each phase with a tiered solution, maximizing the development of the Revit design and geometry.

For the purpose of describing each phase and the possible integration of workflows with other disciplines, responsibilities have been broken down separately into Architectural (A) and Structural (S).

Pre-Design:

- **A:** Often referred to as Preliminary Design (PD), the pre-design phase involves Architectural Design Teams extensively utilizing modeling and drafting tools to address basic programming project requirements. The design teams will consider multiple iterative design concepts and select a final design solution that meets the project requirements.

  Often, the tools utilized for this process incorporate other design packages such as, but not limited to, the following:

  - Google SketchUp
  - Rhino
  - AutoCAD

  Although these tools are appropriate for use early in the design process, a drawback to their use has been the repetition of production effort when the design team moves into the Revit platform. As an alternative, the team can utilize Revit's conceptual modeling tools to develop the design during this phase of the project. The key advantage to this approach is that the conceptual design work produced in Revit can be easily converted to other Revit geometry as the project transitions into subsequent phases.

- **S:** Typically structural engineers have minimal involvement in the PD phase, beyond advising the architects on what is possible from an engineering perspective.

Schematic Design:

- **A:** With the Revit conceptual model in place, during the schematic design (SD) phase, the Architectural team will begin to integrate system and model family components to define a more accurate representation of the Revit geometry. During the SD phase, a level of detail (LOD) accuracy of the model is not as necessary, and the team can begin to utilize generic system family components such as walls, floors, ceilings, roofs, as well as a basic kit of model component family parts such as doors, windows, curtain walls to begin defining a more accurate representation of the building. The relevance of the model geometry should meet the SD phase documentation requirements, and preliminary sheets with annotations defined in the Revit project.

- **S:** During the latter portions of the SD phase, Structural Engineers will take control of the Levels and Grids of a project, and begin sizing the required structural members. The relevance of the model geometry should meet the SD phase documentation requirements, and preliminary sheets with annotations defined in the Revit project

- **A & S:** Begin initial coordination of Revit files.

Design Development:

- **A:** As the Revit project progresses into the Design Development (DD) phase, generic system families and model component families are replaced with those that contain a higher LOD. During this phase, items such as assemblies for walls, floors, roofs can be further specified, and rooms, schedules, and detailed call-outs such as walls sections and enlarged plans can be further developed. At the end of the DD phase, the accuracy of the Revit model geometry in both LOD accuracy as well as the general “I” in BIM is in place.

- **S:** Engineers will continue to further develop the super-structure of the building, also including analytical calculations to validate the integrity of the structure. Further refinements to the LOD of the model are also made, including but not limited to foundations, columns, framing, re-bar, schedules, and call-outs such as wall sections and details. At the end of the DD phase, the accuracy of the Revit model geometry in both LOD accuracy as well as the “I” in BIM is in place.

- **A & S:** Weekly coordination of Revit files among all disciplines involved, including MEP.

Construction Document:

- **A & S:** With the majority of the modeling now complete, during the CD phase both Architecture and Structural disciplines are considered to be in an additive workflow. The model geometry as defined in the DD phase has changed little, but through an additive process, views are created defining the highest LOD required in a set of construction documents. Final considerations are made regarding information contained within the schedules, and the information is reviewed for compliance.

Closing Comments

The workflow as outlined defines a workflow for a Design, Bid, Build type project. In the case of Design Build projects, there will be differences in the Level of Detail requirements for each phase. Having an understanding of the Revit requirements for the Design Build project prior to execution will be paramount for success.

2.0 Revit Project Setup and Templates

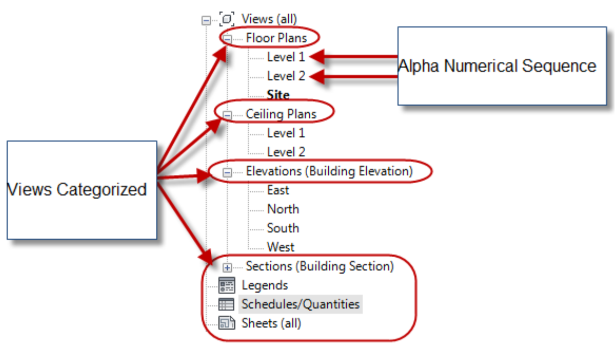
Of important consideration is how a Revit project is started from the configuration standpoint. How the Revit template is set up, and configuration of critical items such as the Project Browser organization, and justification of Revit's coordinate system would allow for ease of user interaction with the file, as well as a consistent understanding of how the files relate spatially.

As Revit project files get larger, and such things as worksharing is enabled, further strategies, some as simple as having a file naming convention, or more complex concepts such as a well defined workset strategy will facilitate a more coordinated Revit project, and set the Revit production team for success.

2.1 Project Browser Organization and Customization

Creating views is the method utilized to derive the appropriate **Plans, Sections, Elevations, Callouts** from the Revit model. You are not limited to a finite number of views that can be created, and therefore it is important to organize the views in the project browser. This becomes increasingly important later in a project where we will have hundreds if not thousands of views that are utilized for multiple purposes such as working views, sheet views, export views, coordination views …etc.

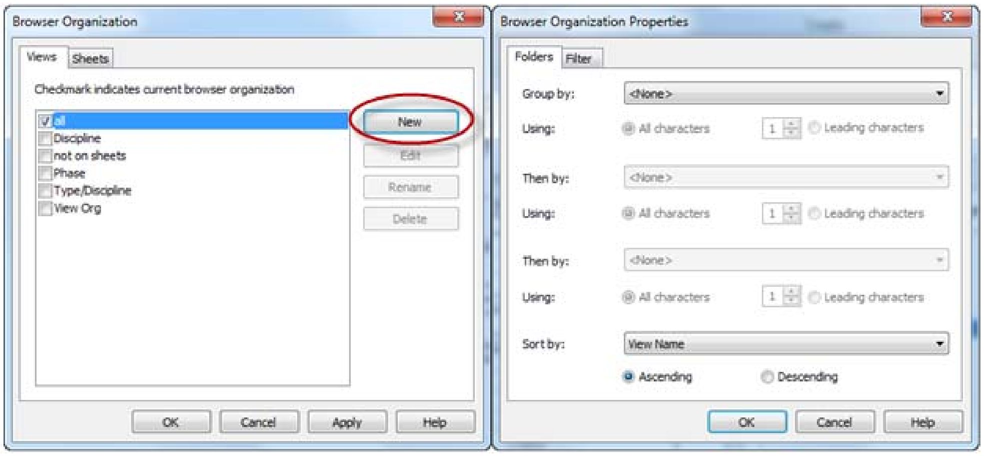
By default Revit organizes views by Category, (i.e. Floor Plans, Ceiling Plans, Elevations, Sections, Legends, Schedules, and Sheets), then alpha numerically.



(Figure 2.1a)

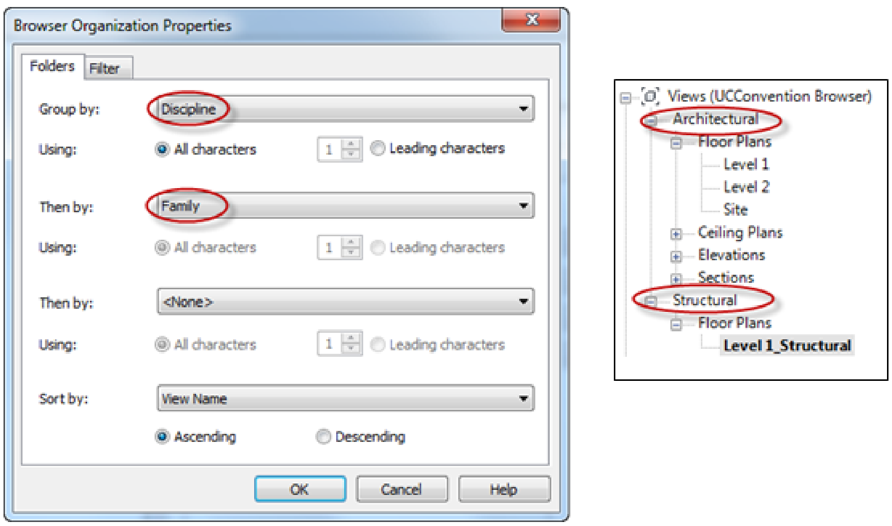
The main problem with this type of sequencing is that the default categories will have way too many views assigned to them. What if you had more than one Level 2 view, what if you had five? How would you be able to differentiate between them?

The Project Browser is organized and sequenced based on parameters that are listed in the view properties. Because of this, you can customize the way the Browser is organized from the **Browser Organization Properties** window. This can be accessed from the **View Tab → User Interface → Browser Organization.**



The Browser Organization Properties
(Figure 2.1b)

By specifying an option in the **"Group by"** parameter, you have the ability to sequence how the Project Browser categorizes and lists the views that you create.



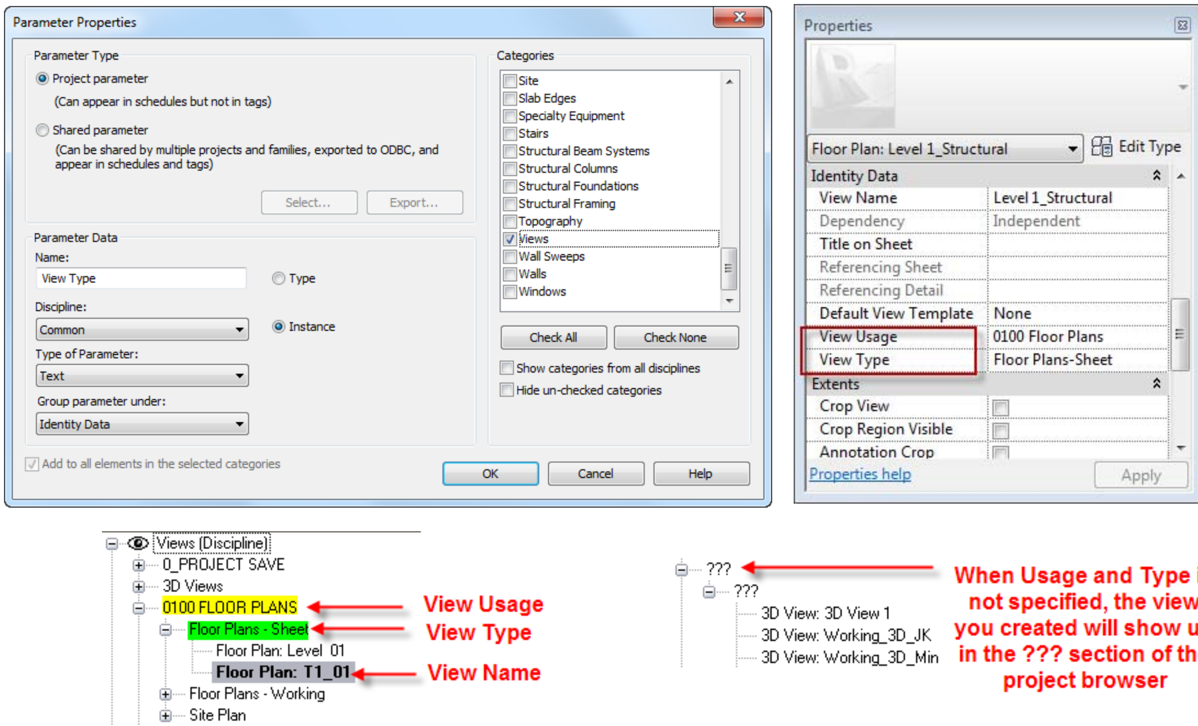
(Figure 2.1c)

In Figure 2.1c you can see by the settings that the view is categorized by Discipline then by the view Family. In this method, you can separate the views in clearly defined Categories, and have the ability to easily navigate the Project Browser

For increased flexibility in the Project Browser, you can create additional parameters for the View Properties. Custom parameters can be added from **Project Parameters** tool listed in the **Manage Tab**.

In the following example (Figure 2.1d), custom instance view parameters were created from the **Project Parameters Window**.

- View Usage
- View Type



(Figure 2.1d)

Both the **View Usage** and **View Type** parameters have pull down menus. All available View Usage and View Type parameters are available to you. Once you specify these parameters, the view will be listed in the appropriate section of the Project Browser.

If you do not specify a **View Usage** and a **View Type** parameter, the view will be listed in the “**???**” section of the **Project Browser.** This is done because the parameters were not assigned to the view.

Important Note: The organization method of the Project Browser should be controlled by the Project Team Lead or the BIM Manager. The project team must be educated on how to categorize and assign views. Just remember a created view does the team no good, if no one can find it.

3.0 Organizing Worksets

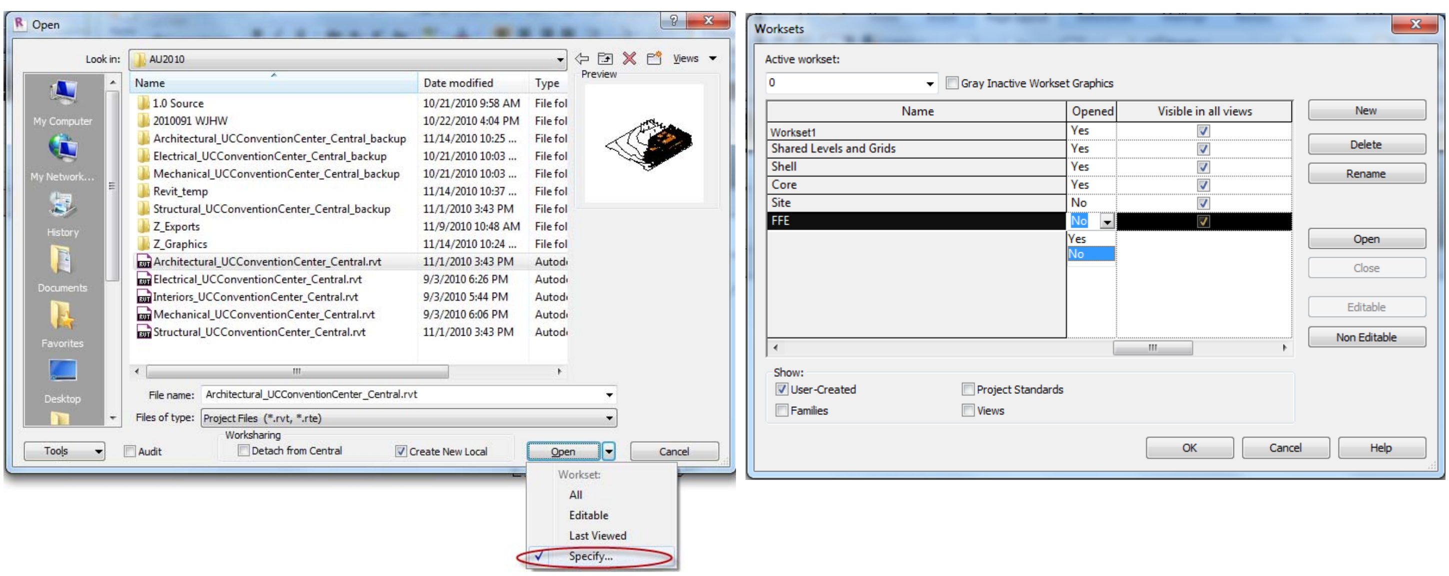
Worksets are utilized in Revit's work-sharing environment. The worksets allow you to divide up the Revit model so that multiple people can work on them. Although many users are tempted, do not think or worksets as layers. Think of Worksets as assemblies and components. In AutoCAD it is perfectly appropriate to have a layer for wall, door, window, but in Revit you may have worksets such as Exterior Envelope, Podium, Core, FFE.

In Large Revit projects, how you separate the model into worksets will be the difference between a stream-lined Revit model and a large clumsy model. Through the effective use of worksets, it provides extreme advantages for the Revit team developing the project. As a note, because of the advantages of worksets, even single user Revit projects are often converted to the work-sharing enabled.

At the time Revit worksharing is enabled, elements that are not part of a standard Revit workset are assigned to a default **Workset1**. After the project workset names are defined, the model objects in **Workset1** are allocated to the new worksets.

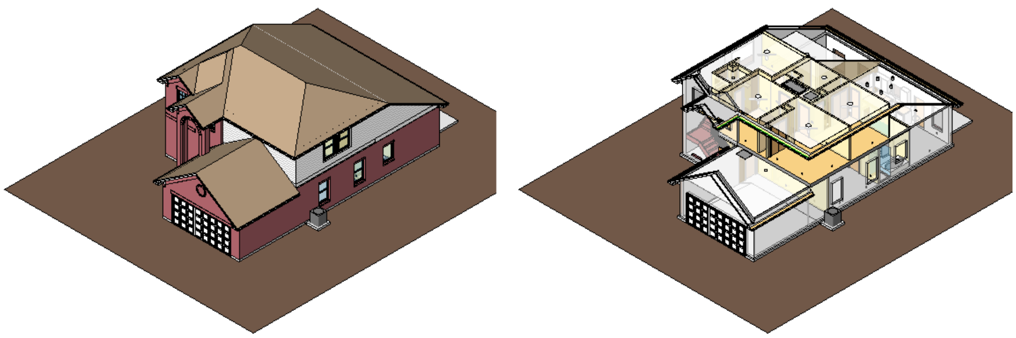
Although you have the ability to rename any workset, it is recommended that you keep the name of **Workset1** as named because this is the one default workset that can never be deleted from the Revit project. **Workset1** can be used as a temporary workset for those unsure of what workset to assign a model object. A 3D view with **Workset1** visible can be used periodically to re-assign these 'temporary' elements to more appropriate worksets.

Effective Workset Strategy: An effective workset strategy for a large Revit project allows the Revit team to selectively filter what information is loaded into the available RAM of the user's workstation. Each workset can be loaded or unloaded from the **Worksets** manager when you start the project.



(Figure 2.2a)

If you choose **No** for a workset, as the Revit file opens it will not load model objects that are assigned to the workset to the physical RAM. In ***Figure 2.2a*** you can see how the worksets for Site and FFE are being turned off at startup. The worksets that are turned off are also specific to the Local file and will not affect other users. Ultimately, by only selectively opening worksets that you need will allow for a lighter Revit file that is more responsive and can be easily manipulated. In the event that you need to see the information from a workset, then it is just a matter of turning it back on.



The Shell workset turned on (left) and turned off (right)

(Figure 2.2b)

Important Note: Just remember to turn all worksets on when you are ready to print. This will ensure that the views you set up for print will be complete and without issue.

Workset Naming: Implement a good workset naming convention. As a best practice, one person on the project team such as the team lead should be responsible for the maintenance and creation of the worksets. Typical team members should not create worksets on their own. If you allow this, you will see your Revit project spin quickly out of control. When new worksets are created, be sure to communicate this to the Revit team. New worksets are of no use if no one utilizes them.

As a sample convention, the naming could follow this convention:

[DISC] – [SYSTEM] – [ZONE]

Examples:

ARCH-Core-Entry	INTR-Demising	HVAC-Equipment
ARCH-Shell-Base	INTR-FFE-Level01	SITE-Planting
ARCH-Rooms-Level01	INTR-Units-Ballroom	

What is important about the naming convention is that it is clear and easy to understand. You want to make sure that the Revit team assigns developed model geometry onto the correct worksets.

Important Note: One of the biggest problems that can occur on a Revit project is when vast quantities of Revit objects assigned to wrong worksets. This causes the entire workset strategy to become in-effective. The first line of defense is educating your Revit project team on the importance of worksets. Defining a clear understandable workset naming convention helps take the guess work away.

4.0 Revit Coordinate Systems

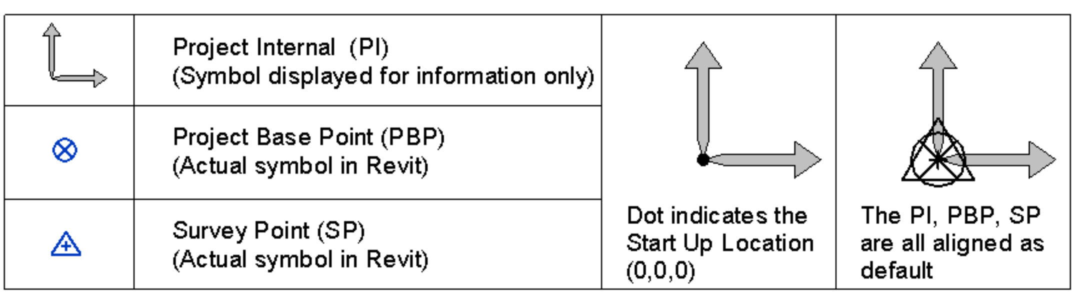
In the process of sharing Revit models on a collaborative project, one of the most important considerations is the basis of a common coordinate system. Agreement on a fixed point that represents the (X=0, Y=0, Z=0) will allow for an easy and consistent means of referencing Revit and CAD data into the host Revit model. The determination of this point as a best practice should be a major part of a BIM project kick off plan. This section will define the components of the Revit coordinate system, and best practices for project configuration.

In Revit there are two main coordinate systems that must be considered, the project internal, and shared. Both have features and limitations. Utilizing the correct coordinate system properly will allow for the easiest collaboration of the Revit models, as well as any additional supporting CAD based files.

Project Internal: Every Revit file has a Project Internal coordinate system referred to as "*Project*". This is referenced in several locations in Revit such as the export settings for CAD files, within the type properties of level datum for the Elevation Base, and type properties of Spot Coordinate objects. (It is strongly recommended that all Revit geometry is created within 1 mile of this point.) The **Project Internal** coordinate system is equivalent to the **WCS** (World Coordinate System) commonly utilized by other 3D applications. Therefore in Revit, the true origin (0, 0, 0) of the Project Internal coordinate system, referred as the **Start Up Location**, is a fixed point in the Revit file that cannot be changed. Also, an associated component of the Project Internal coordinate system is the direction of the Y-axis is representative of **Project North**. This setting is default in all orthogonal views, and recommended that your building geometry has a direct relationship to the Project North, as it would be viewed on a Sheet.

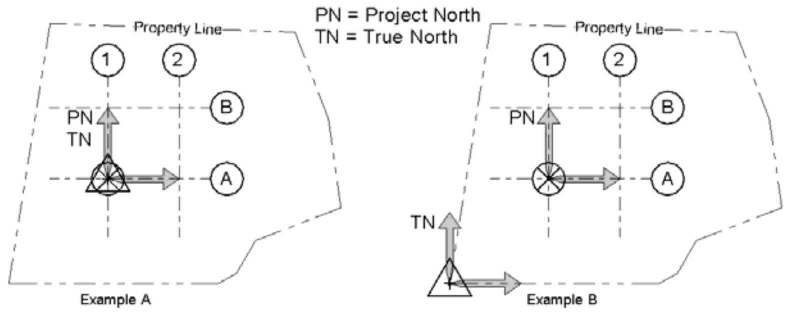
Shared Coordinates: Simply, the Shared Coordinate is a user defined coordinate system that has an origin point and a rotational value that defines the **True North** in relation to the **Project North**. The Shared Coordinate can be synchronized between multiple Revit models, and AutoCAD drawings. Through the Shared Coordinates, it would be possible to specify an alternate coordinate system with an origin that represents the (0,0,0) of a geodetic survey marker, or a station pin.

Important Note: In Revit there is no visible graphic icon that represents the **Project Internal** coordinate system. As a default when you start a new Revit project utilizing the out of the box template, two objects belonging to the Site category, ***Project Base Point***, and ***Survey Point*** coincide with the Project Internal coordinate system. Moving the **Survey Point (SP)** away from **Start Up Location** sets the user defined **Shared Coordinate**.



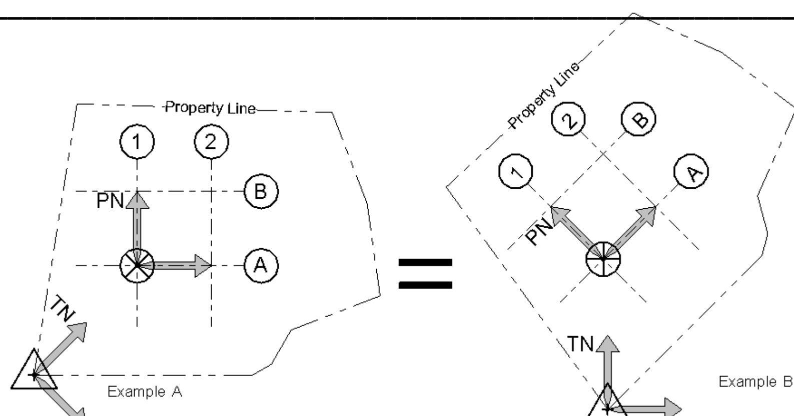
(Figure 4.0)

The following examples show the possible configurations of the **Project Internal** coordinate system, the **Shared Coordinate**, the **Survey Point**, and the **Project Base Point.**



(Figure 4.1)

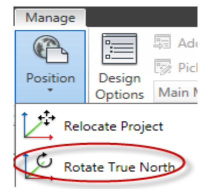
Figure 4.1 Example A	The **Project Internal (PI), Project Base Point (PBP),** and **Survey Point (SP)** all in alignment. The grids were constructed so that the intersections of Grid 1 and Grid A coincide with the **Start Up Location (SUL)** of the **PI**. The direction of the Project North and True North are parallel. (It is strongly recommended that a relationship is made between the grid system and the Project Internal coordinate system.)
Figure 4.1 Example B	The **PI** and **PBP** are in alignment. The grids were constructed so that the intersections of Grid 1 and Grid A coincide with the **SUL** of the **PI.** The **SP** has been relocated to coincide with the bottom left intersection of the property line which creates a user defined **Shared Coordinate**. The direction of the Project North and True North are parallel.



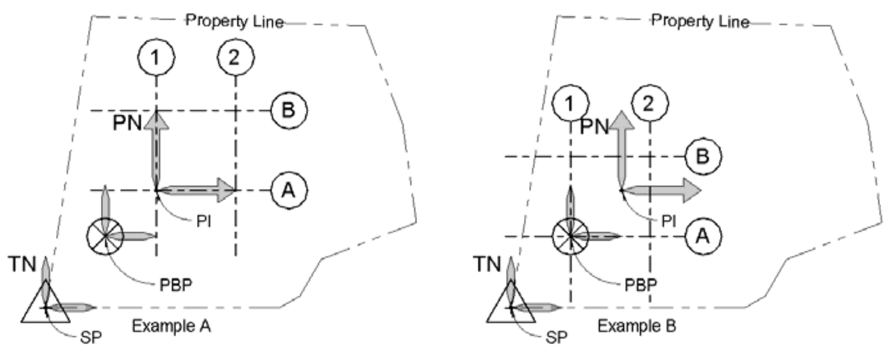
(Figure 4.2)

Figure 4.2 Example A	The **PI** and **PBP** are in alignment. The grids were constructed so that the intersections of Grid 1 and Grid A coincide with the **SUL** of the **PI**. The **SP** has been relocated to coincide with the property line. A rotation of 45 degrees has been added to the survey point which denotes the direction of True North from Project North. The view of Example A has been oriented to be orthogonal with Project North.

Figure 4.2 Example B	The **PI** and **PBP** are in alignment. The grids were constructed so that the intersections of Grid 1 and Grid A coincide with the **SUL** of the **PI**. The **SP** has been relocated to coincide with the property line. A rotation of 45 degrees has been added to the survey point which denotes the direction of True North from Project North. The view of Example B has been oriented to be orthogonal with True North.



<u>Important Note</u>: The examples A and B of Figure 5.2 convey identical information. Only the orientation of Project North and True North are represented differently. In Revit you can rotate the direction of True North from the **Manage Tab → Position → Rotate True North**.

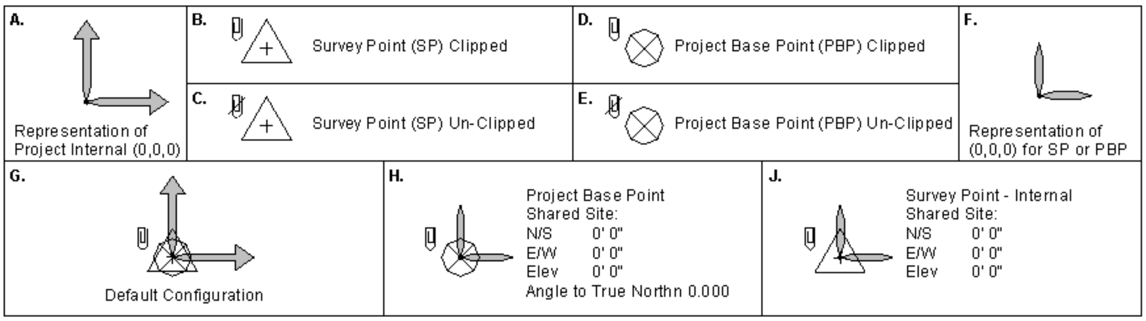


(Figure 4.3)

Figure 4.3 Example A	The intersection of Grid 1 and Grid A is aligned to the **PI**. The **SP** has been relocated to coincide with the bottom left intersection of the property line, which creates a user defined **Shared Coordinate.** The **PBP** has been **un-clipped*** (see following section for **clipped** and **un-clipped** definitions) and moved away from the **PI** creating a new unique (0, 0, 0). In this example there are three independent coordinate systems that have been developed.
Figure 4.3 Example B	The **PBP** has been **un-clipped*** (see following section for **clipped** and **un-clipped** definitions) and moved away from the **PI** creating a new unique (0, 0, 0). The **SP** has been relocated to coincide with the bottom left intersection of the property line, which creates a user defined **Shared Coordinate.**The intersection of Grid 1 and Grid A is aligned to the **PBP**. In this example there are three independent coordinate systems that have been developed. **!!!This is not a recommended configuration!!!**

<u>Important Note</u>: In the case of **Figure 4.3, Examples A and B**. The **PBP** was un-clipped and moved away from the location of the **PI**. The **PBP** in this case creates a new independent (0, 0, 0) The origin defined by the **PBP** is not to be confused with the **Start Up Location (0, 0, 0)** of the **PI.**
Unless your project requires another reference point beyond the Survey Point (SP), it is recommended that the Project Base Point (PBP) always stay in alignment to the Project Internal (PI)

<u>Clipped vs Un-Clipped</u>: In the previous section, we examined the **Project Base Point (PBP)**, and the **Survey Point (SP)**. The location of these objects in relation to the **Project Internal (PI)**, defines additional coordinate systems. In the case of the **PBP** and the **SP**, each object can be in a **Clipped** state and an **Un-Clipped** state.

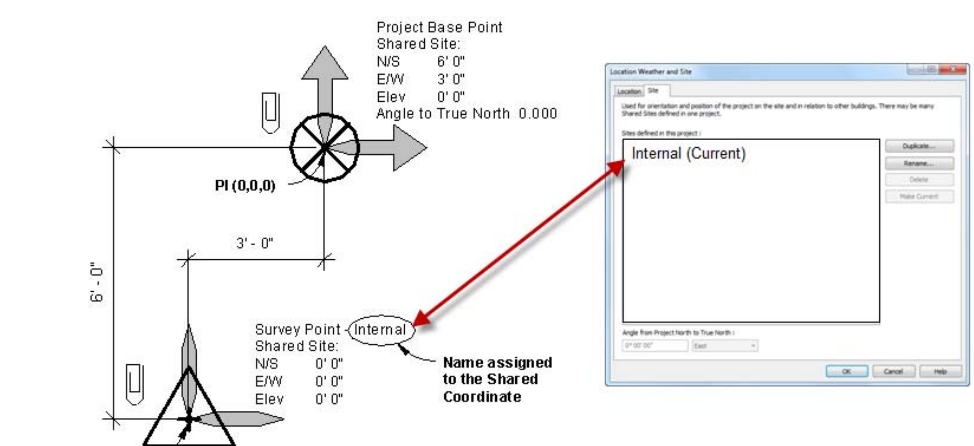


(Figure 4.4)

As a default when starting a new Revit project from the default out of the box template. The **SP** and the **PBP** are aligned to the **PI** (See Figure 4.4, G.). These three coordinate systems are in perfect alignment. To see the PBP and SP graphically in Revit, go to the default Site view, or from any plan view go to the **Visibility Graphics,** go to the **Site** category and check the boxes for **Project Base Point** and **Survey Point.**

The **PBP** and the **SP** can be selected independently. As a default the **PBP** and the **SP** are in a clipped state (See Figure 4.4, B and D). When you independently select the **PBP** or **SP**, it will show you tracking data to the right of the symbol (See Figure 4.4, H and J). The tracking data reports the pertinent (X, Y, Z) of the **PBP** and the **SP**.

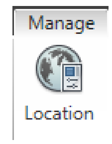
The following examples describe the Clipped and Un-Clipped characteristics of the **Project Base Point (PBP)** and the **Survey Point (SP)**



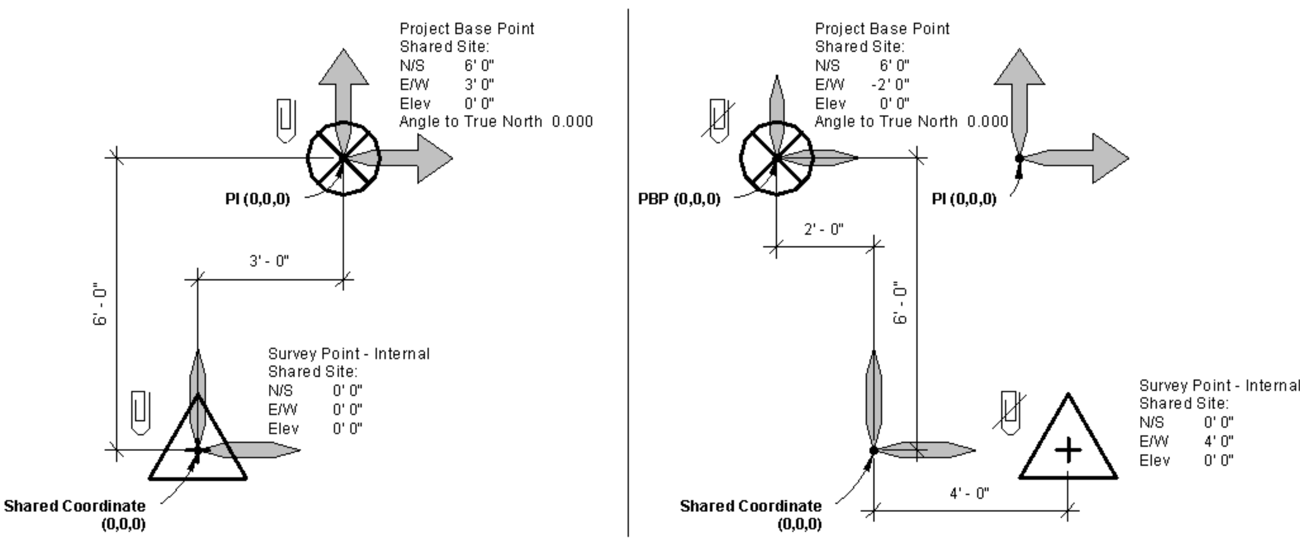
(Figure 4.5)

In **Figure 4.5**, both the **SP** and the **PBP** are Clipped. Note the following characteristics.

- The **PBP** is in alignment with the Project Internal. The tracking data of the PBP will always report the position relative to the Shared Coordinate
- The origin of the **SP** is the same as the (0, 0, 0) of the **Shared Coordinate**. The tracking data is reporting the **SP**'s position relative to the (0, 0, 0) origin of the Shared Coordinate.
- Rather you move the Clipped **PBP**, or you move the Clipped **SP**, the relationship between the two points are identical. i.e. You are performing the same operation.



<u>Important Note</u>: The "Internal" reference to the tracking data of the Survey Point is the default name assigned to the Shared Coordinate. The name of the Shared Coordinate can be updated from the **Location Tool** available in the Manage Tab.



(Figure 4.6)

In **Figure 4.6**, The Survey Point was first relocated in the same manner as described in Figure 4.5. Afterwards the **PBP** and the **SP** were Un-Clipped and moved. Note the following.

- The position of the **Project Internal** and the **Shared Coordinate** remains in the same position.
- The tracking data of the Un-Clipped **PBP** is reporting its position relative to the (0, 0, 0) of the Shared Coordinate
- The tracking data of the Un-Clipped **SP** is reporting its position relative to the (0, 0, 0) of the Shared Coordinate
- The position of the **PI** does not impact the tracking data of the **PBP** nor the **SP.**

<u>Important Note</u>: Unless it is absolutely critical to have an additional point beyond the Shared Coordinate origin, keep the **PBP** aligned to the **Project Internal** origin.

<u>TIP and Trick</u>: Why Un-Clip the **Survey Point (SP)** and move it? You may have a situation where the origin of the Survey Point represents the origin of a Geodetic Survey Marker. This point may be miles away from your Revit Geometry which has a close proximity to the Project Internal. When you Un-clip the Survey Point you do not impact the origin of your Shared Coordinate. You can then move your Un-Clipped Survey Point in close proximity to your project. Major Advantages are as follows:

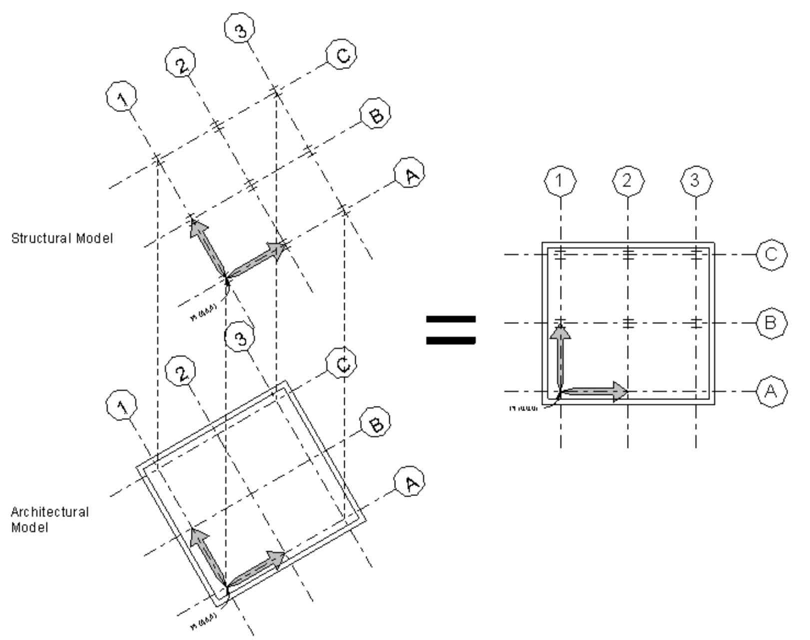
- By having the SP and PBP within close proximity, Revit will not have to compute the miles of virtual space. Revit will be happier from a computational standpoint and
- With the SP and PBP visible, when performing a Zoom Extents. Revit will not zoom out in such a way that items are too small to be visible

5.0 Linking Revit Models

Coordinating Revit models is a very important part of collaboration between disciplines. It is through the linking of Revit models that we are able to share information. In Revit we reference other models into our Revit model by linking them into place.
Section 1.0 discussed the various coordinate systems within Revit. It is with the understanding of the coordinate system that a linking strategy can be devised. When doing any Revit project, there are only two methods that can be seriously considered.

- Linking by Auto - Origin to Origin
- Linking by Auto - Shared Coordinates

Linking by Origin: We looked at the Project Internal (PI) coordinate system which has a true (0, 0, 0) origin. When multiple Revit models are involved, by coordinating the model geometry in relation to the true internal origin allows for the easiest means of overlaying model information. In practice, an example would be utilizing the **PI** to represent the intersection of grids at the lower left limit of a project. Linking models with this origin as a common point of reference will ensure that the Revit project models will be lined up correctly.

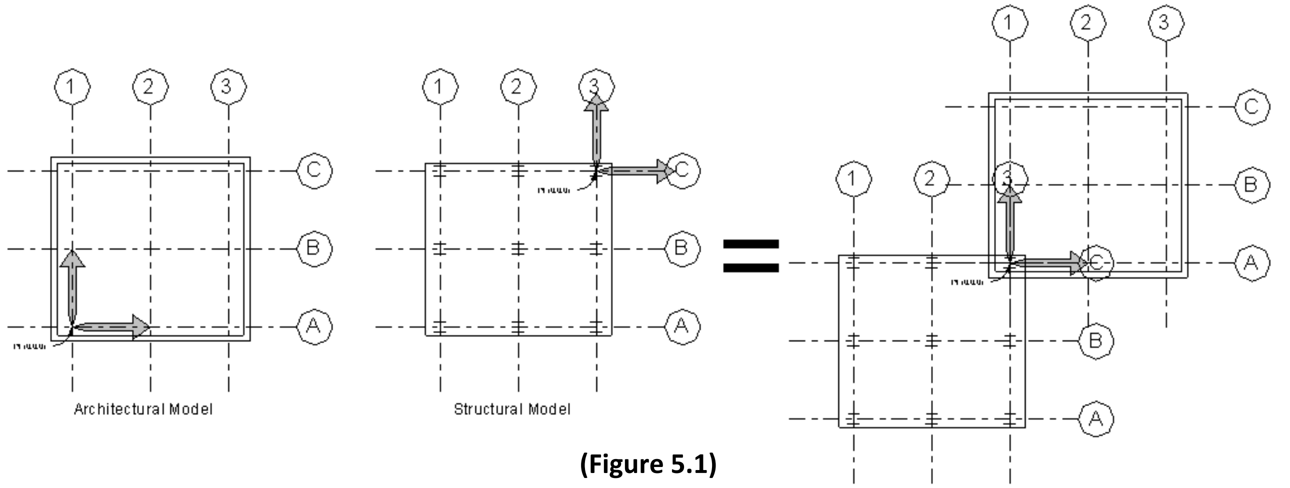


(Figure 5.0)

This method is the easiest and most straightforward method of ensuring the proper alignment of models. Within Revit, the Link Revit window, the Auto - Origin to Origin will automatically align the Project Internal origin of one model with the Project Internal origin of another.

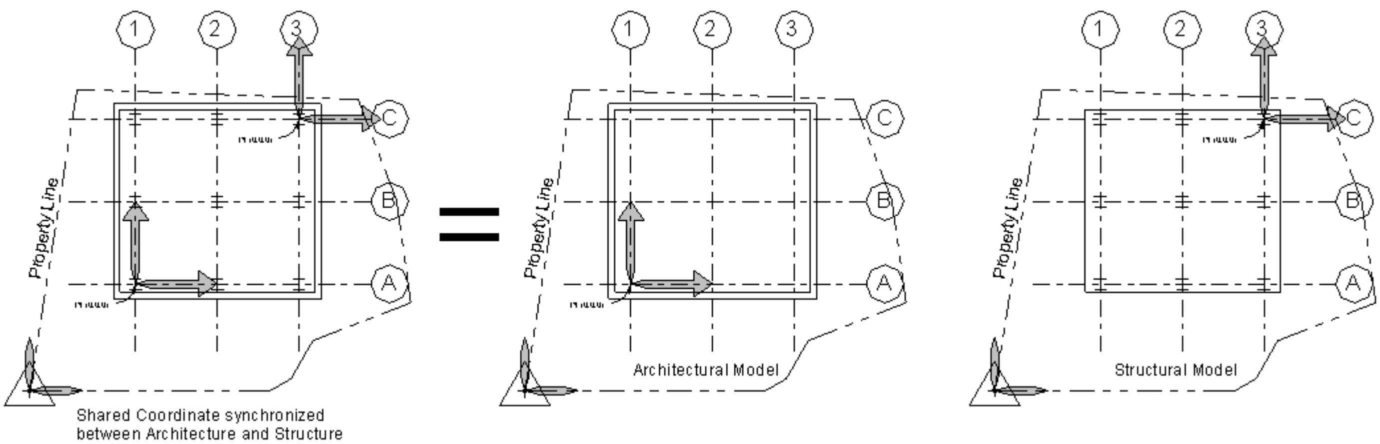
Important Note: The Auto – Origin to Origin only aligns the **PI** of the referenced model to the **PI** of the host model. Do not confuse the **PI** with the **PBP**.

<u>Linking by Shared Coordinate</u>: An alternative means to referencing Revit models is through the use of Shared Coordinates. This method is useful when the Revit models involved have no common origin



(Figure 5.1)

In Figure 5.1, because the Project Internal origin is different for the Architecture model and the Structural model, an alignment based on Auto – Origin to Origin would result in an undesirable alignment of the building geometry.



(Figure 5.2)

In Figure 5.2, the Survey Point of the architecture model was relocated to the bottom left corner of the property line. The Structural model was then referenced in and manually aligned by matching the grid lines. The Survey Points of the two models were synchronized, allowing for a common Shared Coordinate.

<u>Important Note</u>: Developing a Shared coordinate system is an intricate process and must be thoroughly planned out prior to implementation.

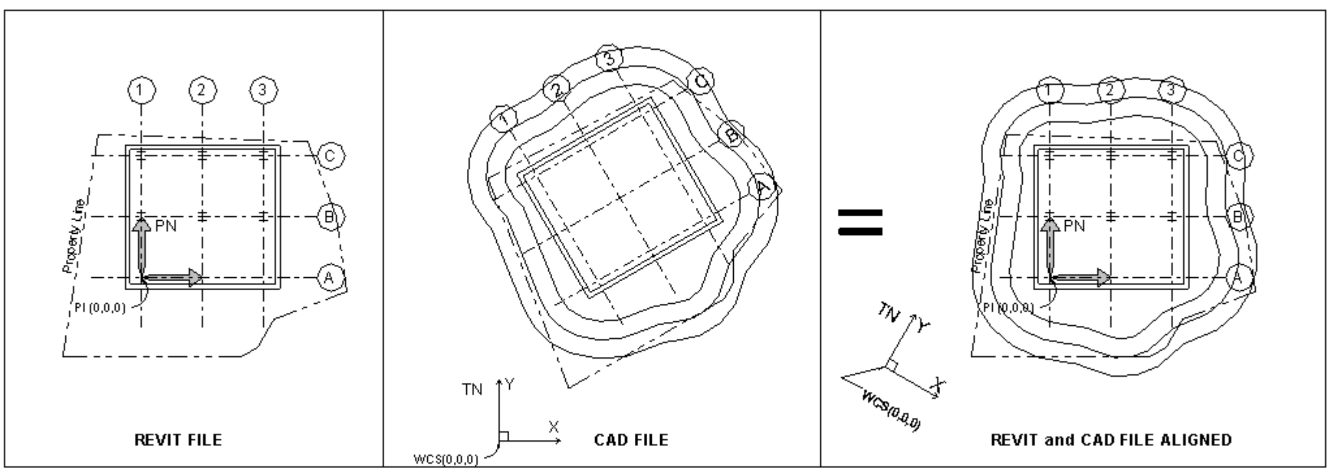
6.0 Linking CAD Files

CAD files can be referenced in to your Revit model for coordination. Similar to Revit files they can be referenced by two primary methods which will help maintain alignment.

- Linking by Auto – Origin to Origin
- Linking by Auto – Shared Coordinates

Linking by Origin: We looked at the **Project Internal (PI)** coordinate system which has a true (0, 0, 0) origin. This point coincides with the **World Coordinate System (WCS)** (0, 0, 0) of CAD based files. If the geometry contained within the CAD files have the same relationship to the WCS, as Revit geometry has a relationship to the PI, the alignment of the two points would bring the geometry together. This would be similar to the condition shown in **Figure 6.0**.

Linking by Shared Coordinate: An alternative means to referencing CAD files is through the use of Shared Coordinates. Initially when CAD files are referenced, only the Auto – Origin to Origin, Auto – Center to Center, and the various Manual placement methods are applicable. Once the CAD file is placed in Revit, align the CAD geometry so that there is a relationship to the Revit geometry. See Figure 3.0

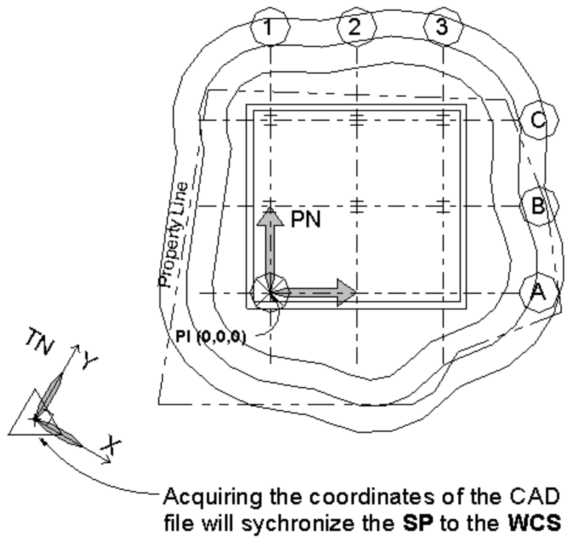


(Figure 6.0)

Important Note: In Figure 6.0, the construction of the CAD file denotes the geometry at a 30 degree angle from the X-axis of the WCS coordinate system. The direction of the Y-axis is representative of the

True North direction.   The alignment of the CAD file was achieved by linking the CAD file into Revit, and then rotated to have the same orthogonal grid conditions as the grids in the Revit file.   By doing this there is clear representation of Project North and True North
Utilize the **Acquire Coordinates** command in Revit and select the CAD file.   This will synchronize the coordinates between the Cad File and the Revit File

Upon Synchronizing the CAD and Revit file, the Revit file will automatically align the Survey Point to the WCS. (See Figure 7.1) You can then synchronize the Shared Coordinate of the Revit file to any other Revit file that is referenced. Once the Shared Coordinate is establish between the files, you can begin linking in your CAD file by Shared Coordinates to any Revit file that has a synchronized Survey Point.



(Figure 6.1)

Best Practice: Civil Engineers will call out the location of the Architects building based off of real coordinates.   The (0, 0, 0) of the engineer's coordinates can therefore be the Survey Point of your Revit file. The advantages of the Survey Point in Sync with the origin utilized by Civil will allow for the easiest collaboration between the Architect and the Civil Engineer. Similarly when exporting Revit data to a CAD format, the export by Shared Coordinates feature will ensure that the exported CAD data will utilize the same coordinate system as the Civil Engineer.

7.0 Effective Collaboration

Copy Monitor

Collaboration Review

Navisworks

This section is briefly covered in the lecture “Connect the Dots: Architectural –Designing the Body”. For more detail on this subject, see the final section of this series, “Connect the Dots: Minding the Whole”