Shared, Project, and Internal Coordinates in Revit

Created by Unknown User (x_shulgan), last modified by Jennifer (Xue) Li on Mar 24, 2020

Major revisions:

- Arkady Gilman, Apr. 2016
- Matt Campbell, Mar. 2011
- Arkady Gilman, Dec. 2005

Coordinate Spaces in Revit

Model Space and Internal Coordinates

Most Elements within a Revit model have some notion of geometric representation and/or location in a common 3D space. This includes elements corresponding to physical objects (walls, furniture, toposurfaces...), imaginary objects (levels, grids, section planes, ...) and even annotations (dimensions, tags...). We will refer to this space as the ***model space***. There is an implicit Cartesian coordinate system in this space that is called ***Revit internal***. When referring to a model space we will often mean its internal coordinate system too.

- Many elements are defined directly in the model space. For such elements (e.g. walls or grids) the coordinates of vertices, edges, etc. in their GReps are always Revit internal coordinates.
- Some elements have their geometry defined in their own, *local* 3D space, which is then mapped into the model space with a conformal transformation (usually non-scaling) specific to the given element. This is often the case for elements that represent physical *instances* of an object. Such element, e.g. representing a piece of furniture, refers to another element, called *symbol*, that contains geometry (and other common data) of the object, that can be shared by many *instances*. In this case geometry of the symbol element is defined in its local space, while the instance element contains a transformation from the local space of the symbol to the model space. In the GRep tree such transformation is part of GInstance node. There may be several levels of indirection like this in a single GRep. In any case, when traversing a GRep and applying all instance transformations to the point coordinates specified in geometric primitives (i.e. GNodes like GFace, GLine, etc.), one will always get Revit internal coordinates in the model space.

All geometric and graphical code in Revit operates with internal coordinates in either model or local space. There are additional spaces and coordinate systems related to views, sheets and output media ("page", monitor) - further info about this can be found in Graphics Coordinates and Transforms.

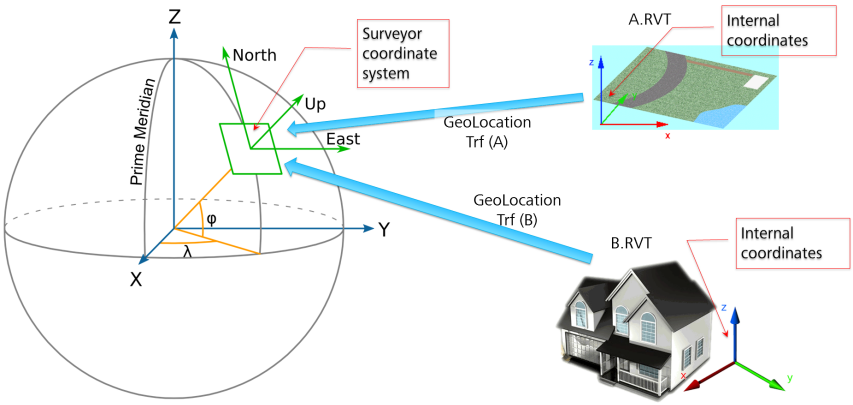
Shared (Survey) Space and Shared Coordinates

Each model (each RVT file) has its own model space. Revit does not assume any relation between model spaces of different models. They are parallel universes. When several models are used in one project, e.g. one representing building shell and another representing structural elements of the same building, the internal coordinates in these two models don't have to be *a priori* "aligned" in any way. Revit makes only two assumptions:

- Z-direction in any model space is always "up";
- All model spaces have the same scale (i.e. 1 unit distance in any model space corresponds to a real 1 foot).

To handle projects consisting of multiple Revit (and non-Revit) models there must be a way to specify location/orientation of these models with respect to one another. For this Revit offers the concept of ***shared space***. This is a 3D space that is globally identified (has GUID) and has Cartesian coordinate system in it called ***shared coordinates***. Revit model contains a mapping from the model space to the shared space (in fact, one model can define several such mappings as will be described later). This mapping is under user control, and it is always a rigid body transformation preserving vertical direction. Once two different models define such mappings into the common (same GUID) shared space, we know how they are spatially related to each other!

Although shared space and shared coordinates is a universal abstract concept, in practice it is typically used to represent *geographic projection* coordinate system also known as ***survey coordinates***. This means that the architects sets the mapping from Revit internal to the shared coordinates in such a way, that readings of the shared coordinates at any point in the model are equal to the ones that the surveyor would measure at these points (see picture). We will use the terms "shared coordinates" and "survey coordinates" interchangeably. The code uses "shared coordinates".



Most of the time surveyors use well known standard "projections" of the portions of Earth surface onto planes, defined differently by zone (e.g. state plane zones in the US). These projections are, in fact, non-linear 3D mappings often involving projection on a cylinder or a cone. They also include the third coordinate (altitude or elevation) in a non-trivial way - different coordinate systems compute elevation relative to different surfaces. The full definition of survey coordinate system includes or implies all the parameters defining such transformation between the Cartesian survey coordinates (x,y,z) and "geographic" coordinates, (latitude, longitude, altitude), often referred to as WGS84. The definition of this transformation is an optional part of the GeoSite element in Revit. Revit does not require this nonlinear transformation to be defined and (at least in R2020) it does not provide a UI to set it up directly or to see its properties. However, if this transformation is defined Revit can use it to transform between internal coordinates and (latitude, longitude).

Implementation details

There is a singleton Element in each Revit model, called GeoSite, that contains GUID identifying surveyor (shared) coordinate system. Site GUID is copied from one model to another when the user asserts that these models share the same surveyor coordinates by acquiring or publishing locations between linked models.

Each model has one ***or several*** GeoLocation elements. GeoLocation is a very simple element that keeps a single translation-rotation transformation (Trf), which is the transformation from shared to internal coordinates. Several GeoLocations may be used to represent several identical units in different places of the city block. Or they may be used to represent different options of locating your building on site. In a sense, GeoLocation is an instance of the whole model. This capability is rarely found in other CAD systems.

One of the GeoLocations is always ***active (current)***. When Revit needs to know survey coordinates of a point (e.g. for annotations such as spot coordinates or spot elevations, or just to show shared coordinates in the UI) it simply takes internal coordinates of the point and applies the transformation from the active GeoLocation element. GeoLocations have user defined names that can be seen in the Location>>Site dialog (along with the indication of the active location).

At any time there is one “current” or "active" Shared Location (can be seen in Manage Locations dialog). The Base Points (see below) allow users to visualize and modify the coordinate systems. Tools like Report Shared Coordinates or annotations like Spot Coordinate tags display coordinates with respect to either the Project CRS (see below) or the active Shared Location. See UI Exposure below for more detail.

When new project is started, it default Project and Shared GeoLocation elements are created with all transformations set to identity.

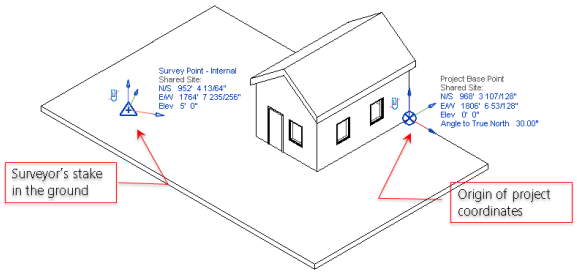
See Shared Coordinates - implementation details for more information, including georeferencing and handling of DWG links.

Project Coordinates

There is yet another coordinate system in Revit called ***project coordinates***. Unlike shared coordinate system that is defined in the shared space, project coordinate system is defined in the same model space as internal coordinates. The Project Coordinate System is used the way an internal coordinate system might be: its origin is generally placed at the building's cornerstone or some other nearby location and it is used to identify the position of items on the building site.As opposed to internal coordinate system that is hidden from the user, project coordinate system is controlled by the user, who can move its origin (he cannot rotate it though - project coordinates are always oriented the same way as internal). Revit does not assume any relation between project coordinates of different models. Revit allows the use of either Project or Shared coordinates in annotations.

Base Points

Every Revit model has two singleton visible elements, one called Project Base Point (PBP) and another called Survey Point (SP). They are implemented as two objects of the same class, BasePoint, having different categories, OST_ProjectBasePoint and OST_SharedBasePoint. They serve as visual manifestations of the project and survey coordinate system respectively and provide in-Canvas UI to manipulate with them. Survey Point have a "clip control".



Project Base Point

PBP simply marks the origin of the project coordinate system. Project coordinates of PBP are always (0,0,0).

- When unclipped, PBP can be moved with respect to the building (i.e. with respect to internal coordinates).
- ~~When clipped, PBP stays fixed with respect to the building and to the internal coordinates, but the entire model moves with respect to the shared (survey) coordinates. This is achieved by appropriate adjustment of the active GeoLocation Trf.~~
  - Disclaimer (Arkady): Moving clipped PBP is equivalent to moving clipped SBP in the opposite direction. Having these two ways for doing the same thing seems to be excessive and confusing.
  - Clip mode is removed in Revit 2020.2. We removed one of these two ways for doing the same thing.

Survey Point

Survey Point (SP) is designed to represent a physical stake in the ground that a surveyor has put or could have put somewhere on the construction site. That is, SP is a point with well known survey (shared) coordinates. It does not necessarily mark the origin of shared coordinates, because in practice such origin is often very far away from the construction site and would be very inconvenient to work with.

- When unclipped, SP can be moved with respect to the shared space and model space at the same time. This means that the mapping from the model to shared does not change (the building "stays put" with respect to the Earth), but the surveyor decided to put his stake in a different place on the ground. The reading of survey coordinates at the SP will change accordingly.
- When clipped SP is moved, this corresponds to moving the Earth underneath the building. Shared coordinates of SP don't change (it is "clipped" to the Earth!), but its location in the model space (its internal coordinates) changes. The Trf in the active GeoLocation is adjusted accordingly.
  - *Disclaimer (Arkady)*: at the time of this writing (Mar. 2016) implementation of SP is deficient - it does not quite implement the "stake in the ground" paradigm when there are several GeoLocations in the model.
  - Linked models that share coordinates with the host model are also interpreted as "clipped" to the Earth during this command, i.e. they keep their position with respect to the Survey Point and, therefore, change their position with respect to the host model geometry (their instance Trf changes, while their GeoLocation stays constant). This is fairly arbitrary interpretation of the user action that is logical when the link(s) represent the site.

Different modifications' effect on Survey Point:

	Clip status	Position	Shared Position	Transform
Specify Coordinates at Point	Clip	Yes	No	Yes
	Unclip	No	Yes	Yes
Modify Survey Point property	Clip	N/A	N/A	N/A
	Unclip	Yes	Yes	No
Drag Survey Point	Clip	Yes	No	Yes
	Unclip	Yes	Yes	No
GeoLocation.setProjectPosition API	Clip	Yes	No	Yes
	Unclip	No	Yes	Yes

Internal Origin

Internal Origin (IO) is the origin of internal coordinate system. Origin of internal coordinate system will become visible starting from Revit 2020.2. Due to its nature, this point is not select-able and modifiable.

UI Exposure of Coordinate Systems

There are several places in the UI where the user interacts with these coordinate systems. For detailed explanation see Revit Help, but conceptually these ways of interaction include:

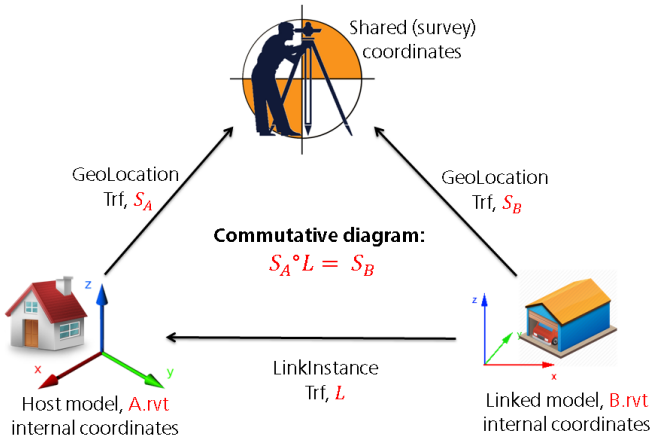
- A tab in the Locations dialog that presents a list of GeoLocations and allows creation, deletion and choosing the "active" one.
- In-Canvas interaction with the base points (see above).
- In-Canvas tools *Report Shared Coordinates* and *Specify Coordinates at a Point* that select model point and allow to see and change shared coordinates at this point and the true north angle. The change effectively applies to the active GeoLocation.
- *Acquire Coordinates* and *Publish Coordinates* commands that apply to a selected linked model (RVT or CAD). These operations results in both models, the host and the linked, starting to refer to the same shared space (i.e. use the same survey coordinate system). The GUID that identifies the shared space is copied from the link to the host (acquire) or from the host to the link (publish). In the latter case the linked model is modified and has to be saved back. When publishing shared coordinates into a linked DWG we create a UCS object in DWG that contains the published GUID and the transformation from DWG internal coordinates to the shared coordinates.
- *Insert/Link* commands (for linking RVT, DWG, point clouds, etc.) have *positioning* options that take coordinates systems into account. Typically there are at least three options:
  - *Center to Center* option means placing the "center" of linked model at the center of the part of the host model that is visible in the current view. This is "safe" option - it guarantees that the link will appear in some "reasonable" place, but does not specify where exactly;
  - *Origin to Origin* option should mean placing the origin of the link at the project origin of the host. Interpretation of the link's "origin" depends on its nature (for RVT link this is the Project Base Point);
  - *By Shared Coordinates* option means placing the linked model in such a way that shared coordinates reported for any point in the linked model would be identical to the shared coordinates of the same point if the linked model were open independently. This is achieved by setting RvtLinkInstance transformation,
    $L = S_A^{-1} \circ S_B$
    where S_A is the transformation from the active GeoLocation of the host model and S_B is the transformation from the active GeoLocation of the linked model. Note that *By Shared Coordinates* option is only available if both the host and the link use the same shared space (same GUID); otherwise the link operation defaults to *Center to Center.*
- Tag-like annotations such as *Spot Coordinates* report either project or shared coordinates of the model point they are attached to, depending on the type properties of the Spot Elevation.

Coordinate Systems and Model Linking

Relation between model spaces of two different models can be created by linking these models. The act of linking creates a pair of elements, RvtLinkInstance and RvtLinkSymbol, in one of the model, which is referred to as "host" or "target". RvtLinkSymbol contains the data about linked model ("external reference") while RvtLinkInstance contains a transformation from the model space of the "linked" (also called "source") model to the model space of the host model. This transformation is represented in terms of internal coordinates of both models. Revit ensures that this transformation is of translation-rotation kind (and rotation is only permitted about a Z-axis which is assumed to be always up).

This is not different from similar concepts in other products, e.g. XREF in AutoCAD. As opposed to many other products, model linking in Revit allows circular references: A.rvt can link B.rvt and B.rvt can link A.rvt at the same time.

As explained above, Revit models can define a mapping from their model space to a shared space even when they are not linked, through their respective GeoLocations. This means that when two models with the same shared space (same GUID in theior GeoSite) are linked, there are two ways of mapping from B.rvt to A.rvt model space (see picture). Revit ensures that these two ways always define the same mapping, that is, that the diagram in the Fig.3 is commutative.



When one of the three transformations (S_A , S_B , L) gets modified by the user, Revit regeneration adjusts to keep the equality $S_A \circ L = S_B$ true. For example, if the user moves the link instance, thus changing instance transformation, L, Revit will adjust the transformation S_B and save it back in the appropriate GeoLocation of B.rvt. Two RegenSteps are used, KeepLocationInSyncWithInstanceStep and KeepInstanceInSyncWithLocationStep.

Handle Large Internal Coordinates

Why are large internal coordinates dangerous?

Although the user is not supposed to care about internal coordinates, things become ugly when internal coordinates of objects in Revit model are huge. Geometrical computations and 3d graphics fall apart. This is true not just for Revit, but for any CAD system. This problem stems from the limited relative accuracy of computations which in turn is caused by the fact that computers operate with limited number of decimal digits (14 in practice) and accumulation of rounding errors.

When Revit computes location of some model feature, e.g. a vertex of a solid, it does it with relative tolerance of 10E-9 (i.e. error in the tenth significant digit). Assume some algorithm needs to check whether two faces of the solid meet at some vertex. The requirement in this case is for the vertex coordinates as computed from each of these faces to be within 10E-4 feet (threshold values are given for example only). Suppose, the solid is located at 10E6 feet from the origin (~200 mi). Then for the first face the vertex coordinate may be evaluated as 1000000.001 while for the second face presumably the same vertex coordinate may be evaluated as 999999.999 (both are within relative error of 10E-9). The distance between the two, however, will be 0.002, i.e. 20x larger than permitted tolerance of 10E-4.

The effects are even more pronounced in graphics, because OpenGL and DirectX operate with only 7 significant digits.

Are large shared coordinates dangerous?

No, they are not. Revit does not do practically any computations with shared coordinates except for reporting them for UI and annotation purposes. Essentially, one of the purposes of Shared Coordinates is to allow such reporting of huge coordinates (as may be required e.g. by documentation standards) while keeping internal coordinates small.

How do projects with huge internal coordinates come up?

In principle, anybody can start new project, scroll the screen some 10000 times (or maybe zoom out 13 times, scroll once and zoom back in 13 times) and start modeling there. In practice this is unlikely, because users are encouraged to start modeling where new project opens its Level 1 plan view.

In the past Revit had certain bugs that allowed "natural" creation of projects with large internal coordinates. To the best of our knowledge such bugs have been fixed long ago.

What if my model is very big?

If you want to model one building in NYC and another in Albany and do it within single RVT file, - this is not appropriate use of Revit, it is not designed for this. Use two separate files instead and link them if needed.

Also if you want to import a map of entire State of New York and then pick location where to build, - this is not appropriate use of Revit too. Clip the map to show appropriate portion and import only that portion.

Other general issues with shared coordinates

1. Shared coordinates code assumes that there is *always* an active shared coordinate. When there is no active shared coordinate, many pieces of functionality will start crashing. However, there are still a few bugs that allow for deletion of the active shared location, most related to synchronizing locations with links. See 170889 and 202938.
2. Users constantly complaining that once the shared coordinate is setup, there's no way to break up the relationship. They need a way to "reset" the shared coordinate of a given file.
3. There are several different representation of a shared coordinate system (ID, WKT or Autodesk coordinate system xml representation). Coordinate system acquired from different approach can have different representation and are considered different coordinate systems when doing compare.

No labels

Shared Coordinates API in Revit

Shared Coordinates - implementation details