



Autodesk Revit – concepts for dealing with large projects

Martin Taurer

LivingRoomCraftZ

livingroomcraftz.com



Goals of this presentation:

- To give an overview of how to structure large projects and utilize work-sharing.
- To show techniques that avoid issues with large work-shared files

Topics:

- Organizing work-sharing
- Model size considerations
- Technical background
- Save to Central
- Central file maintenance
- Audits
- Infrastructure
- Advanced concepts
- When things go wrong

Autodesk Revit – Concepts for dealing with large projects



General

When enabling work sharing on a Revit project file, there are two common practical approaches that are usually recommended:

- Subdividing the project on a syntactical basis – e.g. dividing the project into External Shell, Superstructure, etc...
- Subdividing the project on a spatial basis – work sets for Part A, Part B, Part C,... of the project. Spatial organization of a file though should be utilized only when critical dependencies in the model need to be accounted for. For a spatial organization of independent building parts, file linking is strongly recommend and should be preferred.

Both approaches can be used together, so it is possible to first subdivide spatially and then, based on the spatial organization of work-sets subdividing the project syntactically.



Organizing work sharing

- In a large project it is recommended to critically assess the question whether spatial organization of the project could not be better resolved with File Linking.
- As a general rule it is recommended to assess which parts in the building do exist without direct dependency/relationship to others.
- These parts should be separated into separate linked files linked together in the master file in order to keep the system load low.



Model size considerations

- Large projects require further consideration of common working techniques in Revit. Assessing large project files from a consultant's perspective oftentimes shows a high level of competence of the users in Revit – e.g. features like slab edges, complex nested curtain wall systems and wall curves are utilized.
- However, in a large project, it is recommended to try to keep the complexity of the model low rather than utilizing features of the software to their limits.
- The way the parametric change engine works requires frequent regenerations of element dependencies to keep the model current. The more dependencies that get established in the model, the more complex and time consuming a single regeneration will become. Therefore it is recommended to deliberately keep dependencies low, reducing them to the necessary required to maintain a functional Building Information Model.



Those are specific considerations

Constraints in general:

- Generally speaking, constraints establish relationships within a Building Information Model that have impact on the complexity of the project file and further impact its regeneration cycle. These effects can and will be in effect even when there are relationships between elements residing in different work sets. E.g. the fact that a work set is closed during a session does not mean that a regeneration of a constraint involving an element in that closed work-set will not take place. This is an inevitable requirement for the Building Information Model to work.
- Therefore, constraints should be used prudently. The system itself establishes a large amount of constraints (wall joins, any element hosted by another). In the context of a large project file it is advised to be very conservative when establishing any arbitrary constraint that the Building Information Model could work without. Any click on the blue lock will consequently translate into a constraint that the Building Information Model needs to percolate through to all affected members during a regeneration cycle.

Autodesk Revit – Concepts for dealing with large projects



Floors:

- Floor sketches associated with the surrounding walls are one set of dependencies that can be avoided when trying to optimize a large complex model. In this scenario, a free floor sketch (that though need to be attained when surrounding walls get modified) can help to keep the complexity of the Building Information Model low.
- Large floor plates should be subdivided into smaller sub entities. Basically this should reflect the way the floor gets actually built. Especially when there is a large number of elements joined to the floors (such as walls, structural columns, etc...) it is better to split the floors into several pieces to avoid long and complex element regenerations. E.g. in a case where you have 100 columns joined to one large floor plate, Revit will regenerate all dependencies of all columns when one column is moved. If the floor is split into 3 parts, this regeneration will be cut to a third.
- Floor joins also create element dependencies that can usually be avoided. E.g. rather than having a wall going from ground level to fifth floor and joining it to the floor in order to get the correct graphic representation it is rather recommended having walls go from level to level to avoid those joins.



Walls:

- Wall curves – modifying the profile of a wall should only be utilized in situations where the geometric requirement cannot be achieved by splitting the wall into parts and adjusting their upper and lower boundaries independently. Wall profile curves add complexity to the model and should be avoided wherever possible.
- Wall attachments also establish dependencies in the model that attribute to longer regeneration times. It is for example a good practice to rather avoid attaching wall upper boundaries to other building elements but rather attaching them to dedicated levels.



Curtain Walls:

- Curtain walls are another area - for example, if we take a single wall, 1.8 meters long and 3.5 meters high. For a regular wall, Revit treats this more or less as a single prism with 8 edges. After changing the wall to a curtain wall with a 900/500mm grid we now have:
 - 10 curtain wall panels (80 edges)
 - 15 vertical mullions (120 edges)
 - 12 horizontal mullions (96 edges)
 - That totals in 37 prisms with 296 edges. That means we have 8 times the amount of body geometry and 37 times the amount of edges.
 - This simple calculation shows why it is a good practice to keep curtain walls as simple as possible/complex as minimally required. Trying to isolate curtain walls into dedicated work-sets or linked files is also good practice.



Host sweeps:

- Host sweeps establish model relationships. Their use should be critically reviewed in the context of a large model. It may be more advisable to utilize hosted generic model families for this specific purpose.

Materials:

- Quickly changing materials on the fly can have a significant regeneration time as a consequence. Revit determines graphical appearance especially of joined elements by their materials. Material changes will instantly regenerate though the entire Building Information Model. It is a good rule of practice to contain changes of settings (line weights, materials, object styles,...) to dedicated Building Information Model-maintenance sessions by the model administrator and not let them happen as part of ongoing routine editing of a large building model.



Technical background:

- During work on a large building model you may frequently see the word ‘Regenerating...’ in the Status Bar. A regeneration of the building model means that, due to an editing action in the Revit session, the system now looks through the entire Building Information Model to determine which elements are affected. Then, in the case of a work-shared project, it determines whether this editing action is permissible or whether it violates any restrictions that are imposed due to work sharing – in which case the user will get a warning to either make this work-set editable or issue a borrowing request. Once the system has determined that the edit is allowed, it regenerates the model accordingly and passes all required actions to the elements involved.
- In the case of a work-shared file, this all happens in the local file until it is Saved To Central. A Save To Central means that the local file is transferred to the central file with the system subsequently making a programmatic update of the central file according to the information stored in the local file. Then the system basically re-loads a fresh copy of the central file (including all updates done by other users participating in work-sharing) as the local file.



Save to Central:

- There is no clear rule for the best practice on how frequently users should save to central. Given the fact that Save to Central may require substantial amounts of time to commit, it cannot happen too frequently. However, the longer time passes without a Save to Central, the more the local file and the central file will be out of sync when finally Saved to Central.
- The best practice that can be recommended here is that a Save to Central should happen whenever the Building Information Model was substantially altered due to a specific edit. Basically, when a user edits something in his local file and this edit then triggers a very long regeneration of the local copy, this could be an indicator for a suitable point in time to consider a Save to Central.
- For example; in a scenario where a local user substantially changes the part of the Building Information Model under his ownership by, let's say, updating 60 apartment layouts that resided as groups in his local file, I'd consider a Save to Central.
- The opposite example would be an editing session where a user spent the entire day attaching tags, adding annotation text, etc... Here we can more safely wait with the Save to Central, since the overall Building Information Model was not fundamentally changed.
- Keep in mind though that some benign looking actions can have consequence – so for example when we change the type of a curtain wall panel from a curtain panel schedule view, the system will update all curtain panels in the Building Information Model, and these can be hundreds or even more in the context of a large project.

Autodesk Revit – Concepts for dealing with large projects



Central file maintenance:

- Personally I recommend refreshing central files on a fixed schedule. If time allows, I do recommend doing so at least on a weekly basis, with large and complex files even more frequently.
- Refreshing the central file basically means to open it and save it as a new central file under a different name. That will force all users contributing to the work-shared Building Information Model to grab fresh local files as well. The benefit of this is that you start with a new freshly compacted central file that carries a short history tree (backup to the central file). This will also ensure on a practical level that there will be regular copies of the central file that are known to work.

Audits:

- Auditing the central file on a regular basis is very advised – however, auditing the only copy of a central file that already shows anomalous behavior is definitely not. Revit will attempt to resolve issues in the file during an audit. While in most cases this will address the issue, there is always risk it could make it worse. Making a separate copy of a suspicious central file before performing an audit is highly recommended.

Autodesk Revit – Concepts for dealing with large projects



Infrastructure

Hardware

- Building Information Modeling per se will benefit from any more hardware it can get. Building Information Models can quickly become more complex than mechanical models, by orders of magnitude. For that reason it is good practice to invest in hardware.
- Any critical assessment of wages spent on users waiting for regeneration of a Building Information Model happening in virtual memory (due to insufficient physical RAM in the computer) vs. the cost of the memory upgrade is likely to make the hardware investment a bargain.
- A model of the size < 150 Mb calls for a 64-bit operating system (Windows XP or Vista) with more than 3 Gb of physical RAM installed to avoid working in virtual memory. Any operation that the software will need to execute in virtual RAM will result in a minimum factor of 10 times longer for the regeneration to commit.

Autodesk Revit – Concepts for dealing with large projects



Network

- Network reliability and subsequently performance are the prime candidates for improving work sharing reliability and user experience. Most cases of file corruptions originated in network traffic gone wrong.
- Work sharing by itself relies on network traffic. There are two factors that generally impact the performance and reliability here:
- Overall network speed: it is important to identify any hardware bottleneck in the network when work sharing large models in mission-critical situations. Those bottlenecks are not always very obvious, it can e.g. be the 10-year old network switch that resides somewhere behind a stack of paper.
- Network traffic: even more important than pure network speed, we should make a critical assessment on what happens on the network while we are doing critical Save to Central cycles.

Autodesk Revit – Concepts for dealing with large projects



Examples known to impact reliability and performance are:

Comment: all the examples listed below come from personal observations in user scenarios (Martin Taurer)

- Streaming http sites: admittedly, it is nice to pass the time while waiting for a Save to Central on Friday evening by watching a Youtube video or downloading a new purchase from iTunes. But actually it is a very bad idea to do so, since both examples hog bandwidth on the network required to push and get the Building Information Model to the central file and back from it again.
- Virus scanners and mal-ware blockers: basically everything that monitors network traffic on the intranet will affect performance of a Save to Central
- Software updates: modern OS platforms have the tendency to perform unattended updates of installed software, either for the OS itself or for 3rd party applications (such as Java, etc...) These are downloads that easily exceed hundreds of megabytes and result in network traffic.
- P2P and FTP traffic: file exchange through P2P networks or via FTP while Saving to Central simply should be avoided at any given time. FTP exchange is probably worse, since most FTP clients optimize their performance by multiple simultaneous connections. Then there is little bandwidth left of the network to get the local file back to central.
- Video Chats, IP-telephony: both are pure bandwidth hogs and should never happen during a save to central
- Backup systems: Backup systems usually start after work hours; Friday evening is usually prime time for a backup system to start. It is good advice to audit the in-house backup system against any conflicts with end-of-week Save to Central operations.

Autodesk Revit – Concepts for dealing with large projects



Summarizing - in a scenario where we have to deal with a large Building Information Model at a critical project stage I would advise to think highly conservative when it comes to networks. Ideally, there would be a dedicated and contained sub-network without any external connection. The cleaner the network environment is to deal with the stress of a moving work-shared Building Information Models through the office, the better.

Autodesk Revit – Concepts for dealing with large projects



Advanced concepts:

Hardware:

- An ideal infrastructural environment for a large sized model invites thinking of the required infrastructure as a clean-room environment. That would involve
- Workstations connected to the server hosting the central file via a network that has no connection to the outside (Internet) network whatsoever. That basically means a dedicated Building Information Model network/intranet besides the regular office intranet.
- Workstations that are solely dedicated to the Building Information Model with the least amount of third party applications involved. That basically means that every user ideally would have the Building Information Model workstation and a second computer for daily tasks such as email, being connected to the Internet, etc...
- A clean gateway for all incoming files: anything that comes from the outside world into the Building Information Model (like the DWG file from the kitchen supplier) will pass a virus scan outside the dedicated Building Information Model network in order to abolish the need for virus scanning on the dedicated Building Information Model network.
- Workstations with more than 4Gb physical RAM to avoid unnecessary wait times when the Building Information Model gets regenerated in virtual memory.
- 64-bit operating systems to most effectively address 32-bit memory address space limitations.

Autodesk Revit – Concepts for dealing with large projects



Process:

- Ideally, in-house processes would separate Building Information Model production work from Building Information Model maintenance work. That means, all required changes to the Building Information Model for updating in-house standards, styles, graphic representations etc. that will percolate throughout the entire Building Information Model should be done separately and time-offset from regular editing cycles.
- With a project of this size there is ideally a resource that takes care of nothing else than maintaining the central file and taking care of settings, styles and standards common to the entire file; most preferably at a time when nobody else will work on the file.



When things go wrong:

- The basic rule is – take the central file and all local copies, create a copy and store it. It is always a good practice to have a shadow copy of the state of the file when anything grave has happened. Get all journal files and dump files of the local workstation where the issue has surfaced. Support will greatly benefit from them because they give a first hand clue on what has happened. Think in terms of a clean-room environment.
- If working on a local file and anything unexpected happens along the lines of a very unexpected warning or any anomalous behavior of the Building Information Model, do not Save to Central. If indeed any corruption has happened in a local file, the Save to Central could probably commit it to the central file as well.
- If any warning on any workstation appears that a Save to Central was not successful, immediately inform the administrator of the Building Information Model and stop all activity. Copy the central file, open it and try to resave it as a new central file under a new name/path. Unsuccessful Save to Central operations are the major reason for issues in work sharing environments.
- If Save to Central is successful but the local file behaves unexpectedly afterwards, also immediately inform the administrator of the Building Information Model and stop all activity. Again, copy the central file, open it and try to resave it as a new central file under a new name/path. Assess the file, if it seems to behave normally, audit a copy of the file.

Autodesk Revit – Concepts for dealing with large projects



Thank you
Questions ?

Martin Taurer

livingroomcraftz.com