Best Practices for Revit® Model Management

Revised March 4, 2010

The information contained in this document is time-sensitive as the technology and system requirements continually evolve. While D|C|CADD has made every attempt to validate all information, please verify and refer to the official Autodesk System Requirements posted on http://www.autodesk.com. Unless otherwise noted, all information applies only to the Autodesk 2010 versions.

The Importance of Model Management

In the transition to Building Information Modeling, it is important to realize that the office workflow will change significantly from past technologies—regardless of the specific BIM authoring tool(s) employed. More information is being exchanged at an earlier stage in the design process, and this has a wide-ranging impact on the development of a project. Everything needs to be rethought, from the billing structure for design services to the depth of consultant involvement at the early stages of a project.

As the design team acclimates to a BIM workflow, the adoption of a company strategy for model management will become important. Effective management is required to ensure that the model data remains coordinated, well-structured, and efficient. Secondary benefits include reduced errors; improved consistency between team members; and gains in production efficiency. The intent of this document is to convey best practices for managing a model authored in Autodesk® Revit®.

Best Practices: Before the Project Begins

Project Templates

Project templates should be employed on all projects to ensure consistent project development, functionality and aesthetics. More than one template may be used to separate content per project type (e.g. healthcare, commercial, religious). Templates should include and define the following:

- Annotation Styles
- Dimension Styles
- Drafting Views: Consider creating a standard drafting view for detail development.
- Families: Include the 5-10 most commonly used types for System Families (e.g. Walls). Limit Component Families (e.g. Doors, Windows) to those that are used on nearly every project. For all preloaded content, ensure that parameters have been assigned so that schedules and keynotes will fill themselves out.

- Fonts
- Import/Export Settings
- Legends: Include standardized content (e.g. ADA clearances, project abbreviations).
- Line Styles, Line Weights, and Line Patterns
- Object Styles
- Phasing: Preset phasing categories and graphic overrides if applicable.
- Project Browser Organization: Preset project browser to display information as desired. This may include sorting views to separate working views from documentation or presentation views.
- Schedules: Set up and pre-format schedules for Windows, Doors, etc. so that they will fill themselves out when content is added to the model.
- Sheets: Include examples of sheets that are consistent from project to project (e.g. cover sheet, floor plan, ceiling plan, detail sheets).
- Tags
- Titleblocks: Include common sheet sizes (e.g. A, E1), company logo, etc.
- View Templates

Avoid populating project templates with an excessive number of families that may or may not be useful for every project. Favor a minimal rather than a comprehensive template.[1]

Family Templates
To facilitate consistent development of internal family content, the out-of-the-box family templates may be refined to reflect company standards. Additional templates may be added to reduce the need for in-place families, which are both less efficient[3] and more difficult to migrate between projects.

Content Libraries
In addition to the default Imperial Library, create supplemental libraries to organized downloaded and company-specific content.   These libraries may include:

- Component Families
- Detail Component Families
- Materials
- Profile Families
- System Families (Revit files containing Walls, Floors, etc.)
- Templates

Best Practices: Starting a Project
After creating a new project using the appropriate company template, a basic set of tasks should be performed to provide direction for the design team. These tasks include:

- Project Folder/File Structure:
  - Server: Create a folder structure to organize the locations of the central file, consultants' files, and any associated files (e.g. project-specific families, CAD reference drawings, etc.);
  - Workstations: Organize local files in a consistent manner on users' machines.
- Model Position/Extents:
  - Define the project's position in space by choosing an origin, whether internal or based on survey data, using project and or survey base points.
  - Enter location data to establish sun and shadow information.
  - Define true north and project north.

- o   Add levels and grids to represent the expected scale of the project.
- Model Organization:
  - o   Add worksets as needed;
  - o   Define phases appropriate to the project;
  - o   If known, set up Design Options for the project;
  - o   Adjust browser organization to reflect the needs of the project;
  - o   Create Shared Parameters if applicable; and
  - o   Load the appropriate Keynote file if applicable.
- Model Views: Create appropriate views for the project, including traditional 2D and 3D views of the model data, 2D drafting views, any applicable schedules that were not included in the project template, etc.

Best Practices: Modeling Efficiently

The development of an efficient model will require the design team to strike a balance between modeling complexity and hardware capability. For example, a 500-room hospital or hotel may benefit greatly from the use of Groups in Revit; however, these groups may also carry a significant performance penalty. In general, the following characteristics of a Revit-based application model can affect performance:[1]

- Complex geometry
- Multiple parametric relationships
- Multiple constraints
- Linked files

Further recommendations are provided below. Also refer to Autodesk's Revit *Model Performance Technical Note*[1] for information specific to Revit Structure and Revit MEP.

- Area Schemes: Remove unneeded area schemes to reduce overhead.
- Arrays: Utilize arrays to insert multiple objects in an ordered fashion, but ungroup and disassociate objects once inserted if this relationship is no longer necessary.
- Constraints: Minimize usage of constraints to avoid conflicts.
- Copy / Monitor: The copy monitor tool may be employed to synchronize elements such as levels and grids among the design team. Excessive copy/monitor usage (e.g. all walls) may prove to be inefficient.
- Design Options: Utilize design options to quickly compare alternatives. Once the final design has been determined, delete unused options to reduce model overhead. For large-scale options (e.g. entire sections of a building), use separate models instead of design options.
- Detail Level
  - o   Geometry: Until wall, roof, window, and door types are determined, use the generic versions of these elements, which incorporate less geometry. Unless material use or other types of analysis will be applied to the model, consider that a generic wall may be sufficient for some projects or project areas.[1]
  - o   Scale: Minimize geometric detail that will be invisible at the chosen output scale. The necessary level of detail in a given model can often be conveyed to a team in terms of a commonly understood drawing scale, such as "Provide detail to a ¼" level of detail" or some other commonly employed measure of scale. As much as possible, leverage the project team's understanding of typical 2D drawing conventions to invest the correct level of complexity into the model.[1]

- Detail Lines vs. Model Lines: Detail and model lines may be used to add information that cannot be modeled efficiently. Use detail lines to represent information that appears in only one view and model lines for information that appears in multiple views.
- Family Creation: In much the same fashion as the overall project, families themselves should be modeled efficiently for maximum benefit. For example:
  o Parametric families consume more resources than static families, but they provide flexibility that may reduce the number of families that need to be loaded into a project—finding a balance is key;
  o Face-based families consume less resources than those that cut their hosts;
  o It may be unnecessary to model 3D geometry in all families, especially if the family only appears in one or two orientations (e.g. plan, elevation). However, 3D geometry may actually be more efficient than the extensive use of 2D detail elements. Practice both techniques to optimize families.
- Groups: Use groups to efficiently update repetitive elements. However, it is best to avoid grouping objects when unnecessary. Delete unused groups from the project browser.
- Importing and Linking (AutoCAD DWG files, Revit, Filled Regions, Raster Images): Unload any links that are not used to improve performance. Large projects may benefit from the use of separate models, linked together (see Splitting the Model below).
- Joined Geometry: Limit joined geometry to necessities.
- Levels in Elevation: Avoid using levels to indicate every vertical reference in elevation (e.g. window head and sill elevations); such a technique will require a user to hide irrelevant levels in every elevation and section view. Develop and use Spot Elevation tools to label these items.
- Railings: Model railings to provide adequate detail for general use (elevations, plans), but avoid the temptation to model all details within the railing. Add two-dimensional details where required (sections, details) to fully describe railings.
- Splitting the Model
  o General recommendations, per Autodesk: "Consistent customer practice is to break up a large model into multiple files of about 160 MB each for 32-bit Revit or 200 MB for 64-bit Revit, then link together the resulting project files. Such a procedure will work best if the user can work on one file while the other links are unloaded for a majority of the time. Engineering consumers of architectural models may have to maintain one or more constantly loaded links, which may affect model size estimation and thresholds for those disciplines." [1]
  o Further recommendations per D|C|CADD: The decision to split a model should be made early in the design process, ideally in the Pre-Design or Schematic Design Phase with all consultants aware of this decision and its ramifications. Splitting the model later in the design process (e.g. DD, CD phases) adds significant time and cost. Specific conditions that may trigger a multiple-model project include:
    ▪ Separate buildings (a "campus" project where each building would be a separate model, linked together on a "master site plan" project);
    ▪ Collaboration between geographically disperse project teams (A single building being worked on by offices that are not on the same WAN);
    ▪ Unusually large files (after all other model management practices have failed, because of the particular needs or scope of a project, it may still not be possible to keep the model to a reasonable size)**; and
    ▪ Phased projects where there is a natural and specific split of the design and/or construction.

        **Where file size is the primary factor, model splitting should be considered a last resort and is not to be undertaken lightly. All model management and modeling best practices methods and procedures should be exhausted before model splitting is undertaken. If a firm encounters project types and sizes that require this approach on a consistent basis, we strongly recommend working in a 64-bit environment with appropriate hardware specifications (e.g. minimum of 8 GB RAM).

- Stairs: Model stairs to provide adequate detail for general use (elevations, plans), but avoid the temptation to model all details within the stairs. Add two-dimensional details where required (sections, details) to fully describe stairs.
- View Range: Reduce the view range to the minimum depth necessary to show the desired content.
- Views: Delete views that are unused in the project. In addition, use the Close Hidden button to close views that are not currently being edited.
- Volumes: Setting the Area and Volume Computation option to calculate Areas only may improve performance; however, this reduces much of the analytical functionality of Revit MEP.[1]
- Worksets: The transparent element borrowing process found in Revit reduces the need for the creation of multiple worksets. However, worksets may be used to achieve certain advantages:
  - Efficiency: Workset may be selectively loaded to reduce memory use and improve performance;
  - Security: Access to worksets may be individually controlled;
  - Management: Use of worksets facilitates division of tasks among team members; and
  - Remote Use: Worksets may be employed as a means to complete work offline.

  Although it is possible to create a large number of worksets, doing so will make model management tedious. Please refer to the *Model Performance Technical Note*[1] for suggestions regarding workset creation.
- Worksharing:
  - Save/Synchronize with Central operations should be performed at regular intervals to ensure that members of the design team have accurate information. Saving to Local at the same time records this updated information to the local file.
  - If a user has been away from a project for several days, he/she may have missed significant changes to the model by other users. In these situations, it is best for the user to make a new local file, as it may take a long time for the old local file to synchronize these changes with the central file.
  - Use the Worksharing Monitor utility to track saves to the central file and workset usage, especially near the end of the project.

Best Practices: File Maintenance
The following are recommendations for file maintenance mirroring those provided by a client with significant Revit project experience[2]. For small projects, this file maintenance schedule may be largely unnecessary, while it may need to be accelerated to keep a very large model manageable. Also note that these maintenance procedures may be applied to files received from consultants to ensure optimal performance.

- Periodic File Maintenance (e.g. every sheet set issue): In addition to weekly maintenance tasks, perform the following steps:
  - Audit the central file;
  - Create a backup of the central file;
  - Create a new central file; and
  - Have all users create new local files from this central file.

- Every Two Weeks: In addition to weekly maintenance tasks, perform the following step:
    o Create a backup of the central file.
- Weekly
    o Audit the central file;
    o Perform a Coordination Review, if using Revit to maintain coordination with consultants;
    o Perform an Interference Check, if using Revit to detect collisions;
    o Review and resolve warnings;
    o Delete unused or redundant views;
    o Purge unused elements that will not potentially be used in later phases of the project; and
    o Compact the central and local files.
- Daily
    o Audit local files upon opening them; and
    o Compact local files when closing them for the day.
- Backups and Archives
    o At each issue date, or as required, use the Detach from Central option to create an independent, archive-ready central file.
    o Backups can be generated via the same process, but this is generally unnecessary, as each user's local file can be considered to be a backup file. If a central file becomes corrupt, it is often most efficient to convert the most-recently-saved local file for that project into a new central file, rather than attempting to recover a backup file that may be obsolete.

References
1. Autodesk, Inc. (2009, March). *Revit Architecture 2010, Revit Structure 2010, Revit MEP 2010: Model Performance Technical Note*. Retrieved May 25, 2009 from http://images.autodesk.com/adsk/files/revit_tech_note.pdf.
2. Cone, K. (2007). *Revit Project Management Class Manual.* H.C. Beck, Ltd.
3. Autodesk, Inc. (2009, April 16). *Best practices for families modeling.* Retrieved May 26, 2009 from http://usa.autodesk.com/adsk/servlet/ps/item?siteID=123112&id=13126290&linkID=9243099.