Locking and Worksharing in Revit Server

Understanding the various lock types used by Revit and Revit Server

Introduction

Revit uses a system of permissions and locks to maintain data integrity during multi-user collaboration. Worksharing through Revit Server leverages this system and adds a new lock type to support the maintenance of the Revit Server network. This article discusses each type of lock used in server-based collaboration.

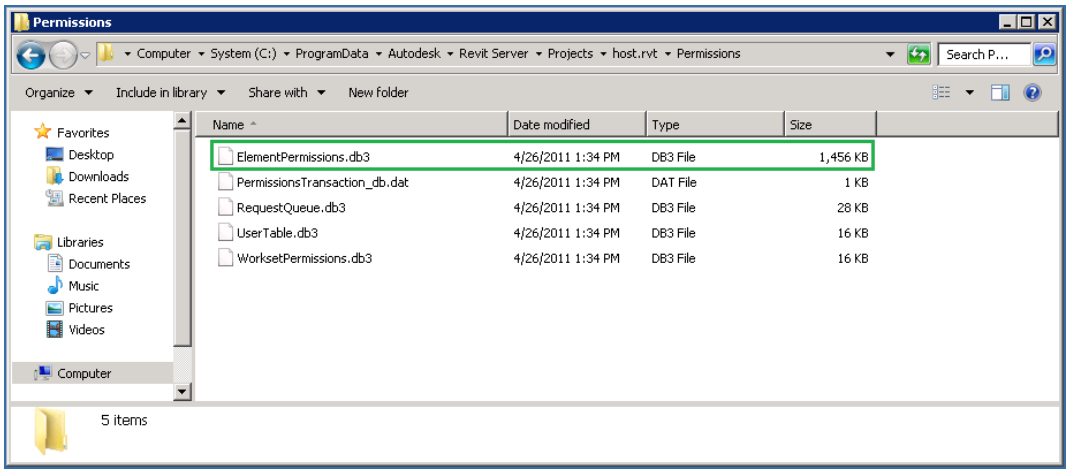
Element-level locks

Element-level locks are employed to control access to each element when multiple users are working in the same model. They are used in both the file-based and server-based workflow and behave according to the following rules:

- An element can only be owned by one user at a time.
- Element locks are associated with Revit users via Revit usernames.
- An element can only be modified by the user who owns it; all other users are locked out.
- A user can only surrender ownership of unmodified elements.
- Changes to modified elements must be committed to the central model or abandoned entirely before ownership can be relinquished.

Element-level locks are acquired by checking out a workset, by explicitly borrowing an element, or by editing an element directly. They can be released when a user performs a Synchronize with CentralWC, and they are always released on the completion of a Relinquish All MineAM operation.

In the server-based workflow, element-level locks are maintained in a collection of SQLite databases. For example, element-level locks acquired without the use of worksets are stored in the file ElementPermissions.db3:



Because element-level locks are associated with users by Revit username (and not by a specific instance of Revit, a particular workstation, or a network user account,) Revit server will treat two individuals with the same Revit username as a single user. In this case, each would be allowed to modify the same element at the same time. This could result in data loss and model corruption, so every user collaborating through Revit Server should be sure to choose a unique Revit username.

In some cases it may be acceptable to assume the identity of another user by intentionally adopting that user's Revit username: If a user must modify an element that is locked by another user, and that user is not available to relinquish it, an administrator can temporarily assume that user's identity and release the lock. Note that this could render that user's local copy irreconcilable with the central model, potentially resulting in the loss of data. Therefore, when feasible, it is preferable for the original creator of the lock to relinquish the element.

Model-level locks

Model-level locks are used in both the file-based and server-based workflows, and they are created by Revit during certain operations to restrict access to the central model when concurrent access could jeopardize data integrity. The locks are automatically released when these same operations complete successfully.

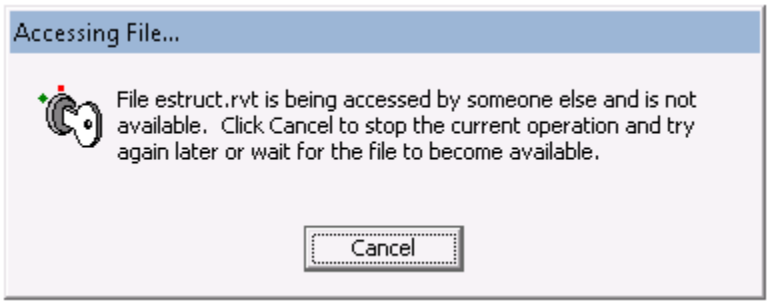
There are two types of model-level lock: read locks and write locks. If a read lock is applied, multiple users are allowed to read data but write operations are blocked to keep the data from changing while being accessed. If a write lock is applied, only one user is allowed to update data, and all read operations are blocked until the changes are fully committed.

There are many scenarios where worksharing operations restrict access to the central model. For example:

- When one user is updating the central model, other users must be kept from reading or writing this model at the same time.
- The central model cannot be updated while a user is creating a local copy. However, other read-only operations are still safely permissible.
- If a user opens a model containing several linked RVT files, those linked RVTs cannot change while the links are being resolved and loaded.
- For the server-based workflow, when a local server's Autosync service updates a model's cache, it retrieves a manifest of all files belonging to that model from the central server. The model's state cannot change while this manifest is being generated.

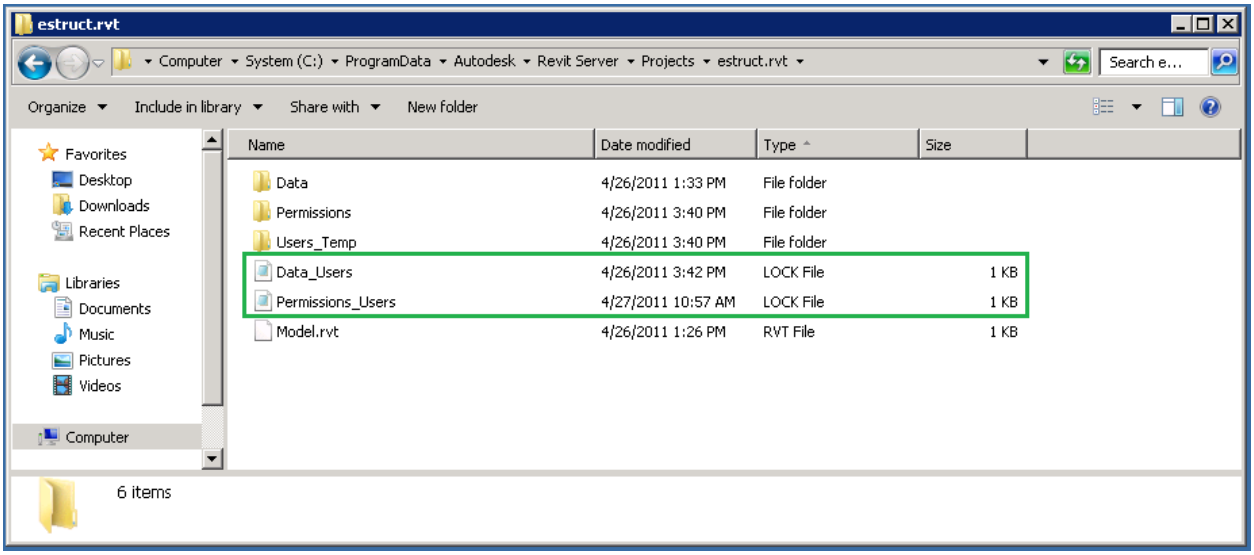
If the appropriate restrictions are not enforced Revit could behave in an unexpected fashion, or worse yet, data could be corrupted.

A model-level lock can be applied to a central model in three ways: by locking its permissions data, by locking its model data, or by locking both. To the end-user these are indistinguishable; whenever any model-level lock is present, Revit simply blocks access to the central and displays the following dialog:



(The cancel button aborts the blocked operation, giving the user the option of working on other things and retrying later.)

In the server-based workflow, model-level locks are implemented using two files in the model's root folder on the central server:



Typically these are both zero in size. When a model-level lock is applied, an entry is made in the corresponding lock file. When the lock is released, the entry is cleared, restoring the file to its original state. The lock entries are encoded in plain text, and the first and third fields can be of diagnostic interest, as they indicate the Revit user who created the lock and the time when the lock was created. For example, the lock file below indicates that an individual with the Revit username "leemic" performed an operation that resulted in the creation of a model-level lock at 8:13:05 AM on April 26, 2011.

Since model-level locks are created and released as needed, they rarely require manual intervention. But if Revit crashes or the network fails while an operation is in progress, a lock could be "orphaned," as the operation that

created it will not have had the chance to release it. In this scenario access to the central would be blocked for all users save the one who created the lock.

Users should always retry any operation that fails to complete successfully, as an orphaned lock will be cleared if a subsequent attempt at the operation succeeds. Instituting this as a best practice will reduce the frequency of occurrence of orphaned locks, improving model availability and user productivity. In cases where orphaned locks persist without evident cause and access to the central model is blocked for an unacceptable period of time, it is also possible to manually release the lock. To do this:

1. Navigate to the central model's model folder and copy the two lock files to a temporary area.
2. Determine which lock has an entry, and view that file with a text editor.
3. Verify that the lock has been in place for more than ten minutes.
4. Identify the user who created the lock, and on that user's workstation locate the journal corresponding to the time when the lock was created.
5. Confirm that the operation that created the lock is no longer in progress, or shut down that user's Revit session entirely.
6. Copy this journal to a temporary area, and send it and the two lock files to Autodesk product support for further analysis.
7. Delete the lock files from the central model's model folder.

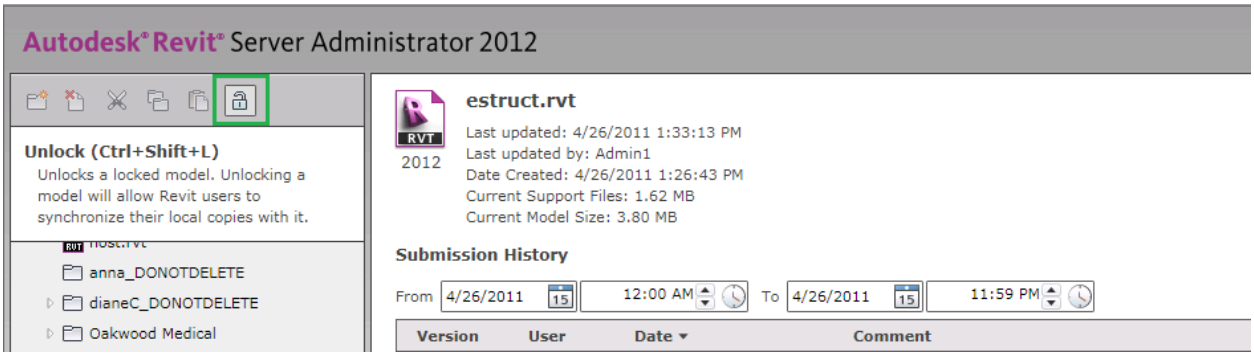
Revit Server will then release the model-level lock, restoring access to the central model for all users. (Although lock files can be viewed in a standard text editor, they should never be manually edited.)

Be aware that manually clearing a model-level lock is risky: model-level locks are created when restricted access to the central model is required to protect data integrity. Releasing such a lock prematurely could allow two or more operations to proceed simultaneously when they should not be allowed to coexist, resulting in data corruption. Before clearing a model-level lock, be absolutely certain that the operation that created it is not still in progress.

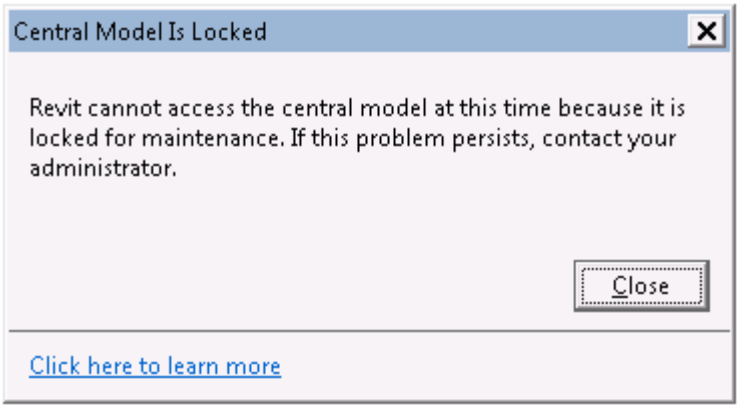
Administrator-level locks

The administrator-level lock applies only to server-based worksharing, as it is used to facilitate server maintenance operations where activity in a model (or a collection of models) should be halted. For example, a model's state should be frozen before the model is archived. Failure to do so could result in an archived copy containing partly committed changes from an update, and this copy would be unusable. If an administrator-level lock is applied first, the model state is guaranteed to be quiescent, and this pitfall can be avoided.

An administrator-level lock is applied using the Revit Server Administrator web-based management utility (or its command-line tool counterpart,) and it can be applied to a single model or multiple models simultaneously.

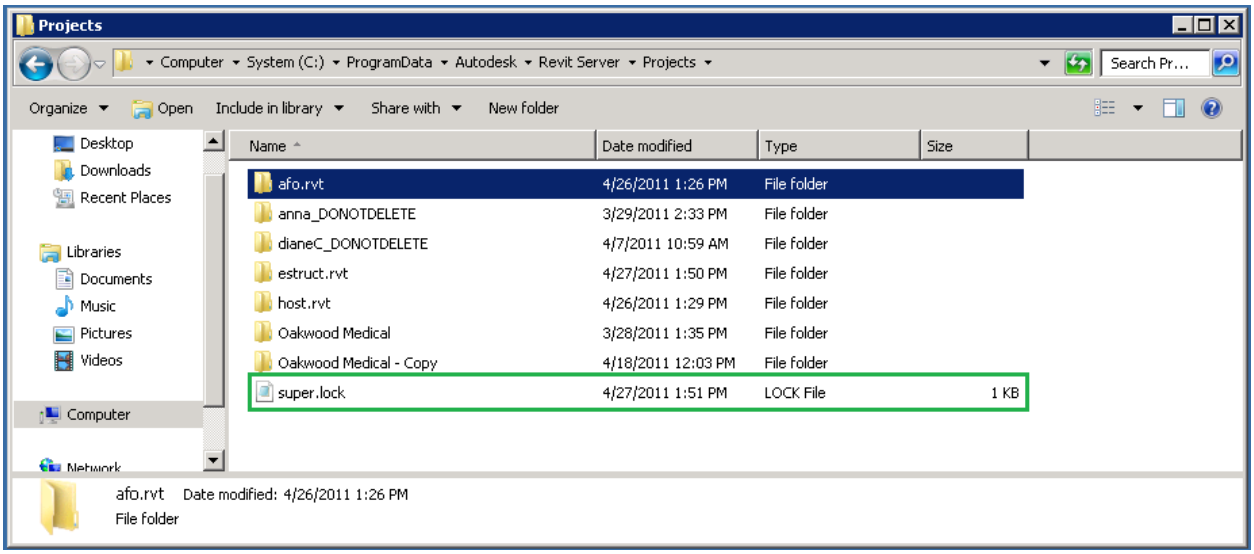


Once an administrator-level lock has been applied, Revit blocks any operation requiring access to the central model(s) with the following message:



The lock is actually applied in two stages. First, all in-progress operations are allowed to continue while new ones are blocked. When the in-progress operations have all completed, the lock is fully committed and further access is unconditionally blocked until the lock is released. One might envision a supermarket checkout line as a conceptual analogue for this behavior: If a checkout clerk turns off the light in their checkout lane, customers already in line can continue to check out, but additional customers are blocked and must go elsewhere until the lane is reopened.

Administrator-level locks rarely require any kind of manipulation. However, if there are any malformed central models (perhaps because a first-time save of a new central model failed) Revit Server may not be able to properly create or release an administrator-level lock. In this case, the lock can be removed by deleting the super.lock file from the central server's root project storage directory. (Note that the offending malformed model should also be removed to prevent the problem from reoccurring.)



Unlike with model-level locks, deleting this file is relatively safe and poses no risk to the live data stored on the server.

Conclusion

Three different types of lock are employed in server-based worksharing. Two are needed to ensure proper operation during concurrent access to a central model:

- element-level locks, which restrict access such that an element can only be modified by one user at a time.
- model-level locks, which restrict access to (and therefore protect) model data during specific stages of certain operations.

The third lock type, the administrator-level lock, is used to facilitate server and server data maintenance.

Locks are intended to be transparently managed and maintained by the system. Observing a few guidelines can reduce the incidence of cases where intervention is required:

- Ensure all users have unique Revit usernames.
- Do not terminate Revit while it is in the middle of an operation.
- If an operation fails, immediately retry it (after restarting Revit if necessary.)
- If first-time save of a new central fails, alert the administrator so that the malformed model can be removed from the system.

These simple guidelines can help ensure a positive user experience when collaborating across geographically dispersed sites using Revit Server.