Use of Revit® Server on High Latency Networks

Recommendations for users wishing to implement Revit Server on a project that spans continents

Introduction

Revit Server is primarily intended to facilitate collaboration where the latency between any two points does not exceed 100ms. This generally constrains use to sites located on the same continent. However, some may wish to collaborate across continents.

To promote the best possible user experience when using Revit Server on a very high latency network, this document discusses the following functions and workflows:

- Element Borrowing
- Synchronize with Central
- Reload Latest
- Creating Local Files
- Bypassing Local Servers

Element Borrowing

When editing a model, users must have exclusive ownership of each element they wish to change. So, Revit will attempt to transparently grant ownership for any element that a user edits but does not own. This must occur nearly instantly to avoid lags and delays during model manipulation. Most perceive an operation that completes within 200ms as "instantaneous." Therefore, in this workflow, a user must be able to acquire ownership of an element in less than 200ms.

Although model data is replicated between the Central Server and each Local Server, permissions data resides exclusively on the Central Server. This keeps a single, consistent view of permissions state, minimizing the opportunity for model corruption. But this also means that all operations requiring access to permissions state – borrowing elements, relinquishing elements, and managing worksets – are transacted directly with the Central Server, even if Revit is connected to a Local Server.

Internal tests indicate that if the latency between Revit and the Central Server exceeds ~100ms, element permissions cannot be granted in less than 200ms. That is because approximately 90ms are used for processing, leaving the balance of the time available for network transport. Hardware compression technologies do not change this calculation, as they only reduce the volume of data moved across the network, not the minimum time required to get from one point to another. As a result, this workflow is not advised for users at sites with high latencies to the Central Server. Any attempt to do so will result in a noticeable lag in element borrowing, degrading the user experience. Instead, these users should take ownership of elements ***before*** attempting to modify them – either through explicit borrowing or by checking out worksets.

Synchronize with Central

When changes in a local copy are synchronized with the central model, the changes must be committed at the Central Server for the operation to complete successfully. For this reason, users working from a high latency site should expect Synchronize With Central (SWC) to take significantly more time than it would from a low latency site. These users may find it helpful to synchronize at regular and frequent intervals, as this will have the effect of reducing the size of the data moved to the central. While the aggregate time spent synchronizing over a day's work may not change substantially, frequent small delays may be more tolerable than less frequent but larger ones. Finally, network compression can further reduce traffic and should be employed whenever possible.

Since slow SWC operations can result in protracted periods of locking for the central model, users at sites distant from the Central Server may indirectly cause delays for users at sites close to it. To mitigate this behavior we recommend that users install the Bluestreak desktop client and the Bluestreak plug-in for Revit. These two components will provide a system of pop-up alerts notifying users when the central model is unavailable due to SWC activity. In this fashion, users can schedule their access to the central file, only attempting to synchronize when the central is free and minimizing the effects of longer and more frequent locks.

Reload Latest

Users at high latency sites may experience unpredictable performance during Reload Latest. When Local Server caches are up-to-date, Reload Latest will generally be very fast. However, when Local Server caches are out-of-date, model data must be pulled from the Central Server and Reload Latest performance may be slower. To maximize the probability that caches remain up-to-date, we recommend securing abundant bandwidth between each Local Server and the Central Server, and using network traffic compression to reduce data traffic. We also recommend periodically clearing each cache of models no longer in active use at that site, by deleting the folders corresponding to these models from the Local Server's cache directory. This will reduce the workload for the Local Server autosync service, maximizing resources for those models that are still under development.

Creating Local Files

When a user creates a new local copy of a model, the model's permissions data must be downloaded from the Central Server. This can be time consuming over high latency networks, particularly when models are large and complex. The **"RevitServerTool"** command line utility is recommended as a scheduled service to automatically create a local copy during off hours. This will shield users from delays incurred by downloading permissions data from the Central Server.

Bypassing Local Servers

Tests have revealed that, in some cases, administrators need not install a (local) Revit Server at each site. Instead, users can connect directly to the Central Server, or they can connect to a Local Server at a nearby site. Those who choose to do this generally do so as a cost saving measure, willfully accepting a tradeoff in performance.

This approach is not recommended for sites with high latency to the Central Server, because users at those sites may be required to shoulder the burden of certain unavoidable data transfer costs. Consciously adding to these costs may further impair productivity. For each site operating at high latency, the usage of a dedicated Local Server is recommended, preferably in conjunction with some kind of network traffic compression technology.

Conclusion

When Revit Server is deployed over a WAN where latencies exceed 100ms, each high-latency site should be given its own Local Server connected to the Central Server via a high-bandwidth connection. We also highly recommend

the use of hardware-based network traffic compression to reduce the volume of traffic that must be moved over a potentially slow network. While we have found that the degree of compression achieved can be somewhat variable, we have also found the value of any compression is magnified over slower networks.

Since creating a new local file can be time consuming over a high-latency network, administrators should configure a scheduled service to automate the creation of the local file. This will hide the delay from the end-user, and it will also serve to refresh the cache at each Local Server. Administrators should also encourage the use of the Bluestreak desktop client application in conjunction with the Bluestreak plug-in for Revit. This will help users work around the increased model contention caused by slower Sync With Central operations.

Finally, we recommend that users operating at high latency sites use worksets or explicit borrowing before engaging in model manipulation, and that they synchronize their changes with the central file more frequently than they might when collaborating on a faster network.