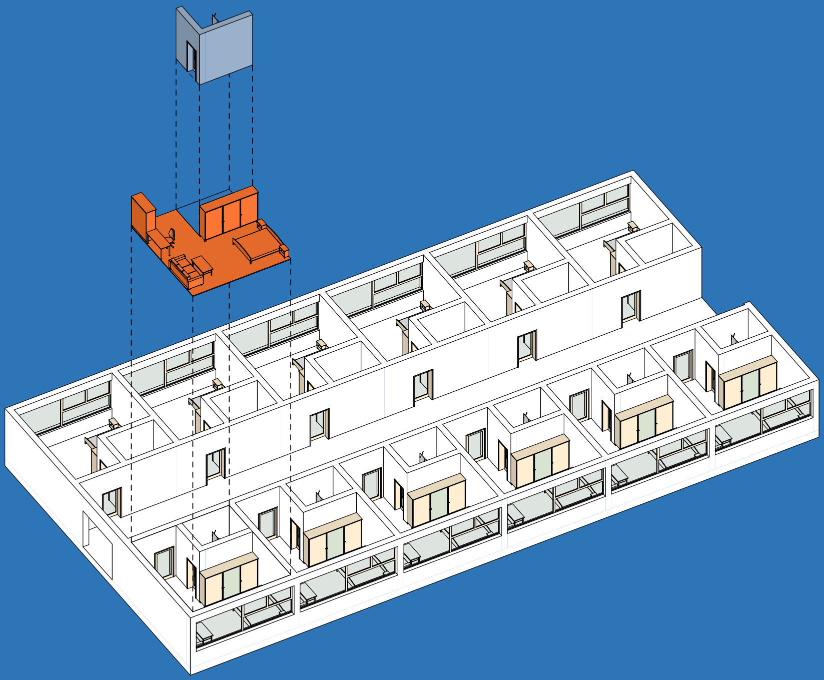
GRAPHISOFT WORKFLOW GUIDE SERIES

Hotlink Management

Workflow examples

Customer Support Services Department

March 2018



GRAPHISOFT
A NEMETSCHEK COMPANY

Exclusively for SSA Customer Use

The Workflow Guide Series are know-how documents providing solutions recommended for BIM workflows and project management related challenges.

The Hotlink Management guide is offering a detailed overview of the hotlink module concept in ARCHICAD and its strategic use for project management.

This document was created with the aim to support the efficiency of your work. If you have any feedback, please send it to helpcenter@graphisoft.com.

GRAPHISOFT
A NEMETSCHEK COMPANY

Visit the GRAPHISOFT website at www.graphisoft.com for local distributor and product availability information.

Workflow Guide Series
Hotlink Management

(International English Version)

Trademarks

Credits

Til Breton, Tibor Szolnoki, Chidambaram Chidambaram, Anna Mindi Coulthard,
Edward Brown, Gordana Radonić

Table of Contents

This workflow guide explains the methods advanced users have developed to maintain large, repetitive, or specially designed parts of the BIM model in the form of hotlinks. We also discuss the effects on the performance for the techniques that are included in this paper. The document's scope of hotlinks in a BIM model is a complex topic suited for BIM Managers though in this document we will try to make it accessible to those with less experience.

1. About Hotlink

1.1 Hotlink – conceptually

ARCHICAD provides two basic methods for sharing/dividing BIM project data among a team of architects:

- **Teamwork** method for sharing projects,
- **Hotlink** method for dividing complex projects into smaller parts.

These two methods are most effectively used in combination to distribute smaller model entities intelligently and make large projects available to a diverse group of participants.

The hotlink method is based on linking external files into one project. It allows us to develop and store parts of the main project file called the **host** in a separate external file, called the source hotlink file, or simply **hotlink**.

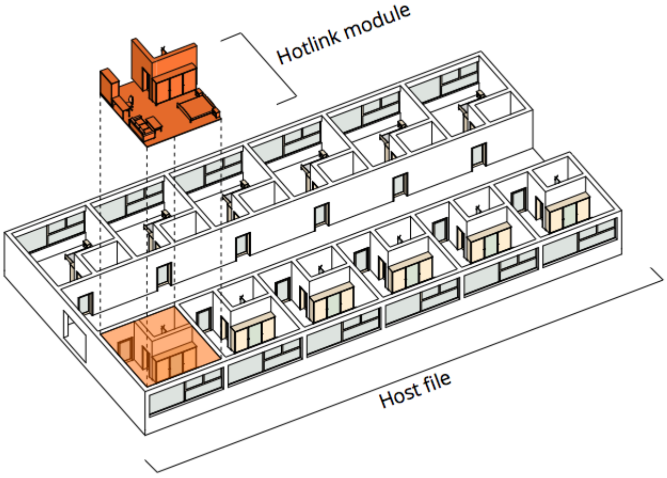


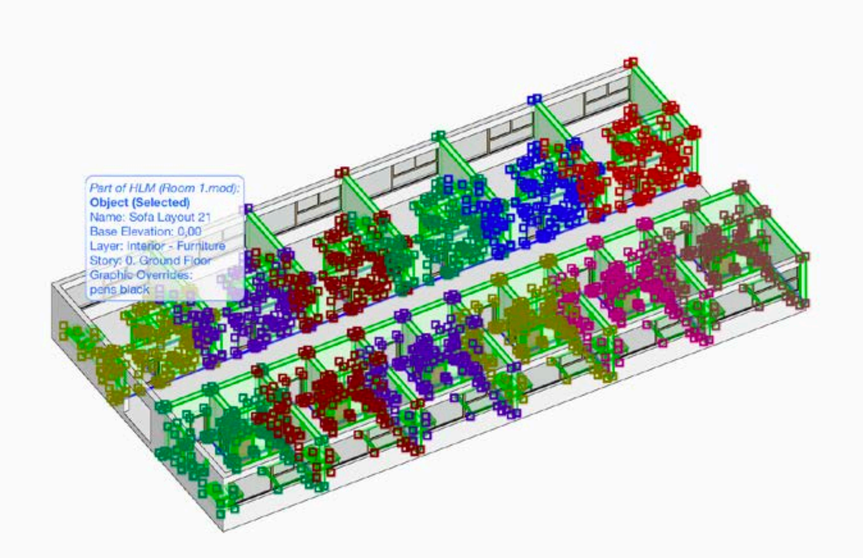
Image 1. - Hotlink concept

The host includes the hotlink source data and a reference to the content of the source file. After we make modifications to the content of the hotlink, the host can be updated.

Read Help Center article [1] About Hotlink Modules.

A host often contains several instances of the same linked content. Repetitive structures of buildings such as hotels or offices with a large number of identical rooms is a classic case. We save a room as a hotlink and place it into a host multiple times. Any changes we make to the room, we make them in the source file. After we update the hotlink in the host file the modification of the one source room will be present in all instances of the hotlink inside the host. Moreover, we can use the same hotlink in multiple projects, which can become an efficient method of re-using the effort and knowledge we put into its creation. Sometimes it is helpful to consider a hotlink as a storage-unit of knowledge.

The Info Tag distinguishes selected elements belonging to a hotlink. This visually reaffirms the fact that a placed hotlink can only be manipulated as an entity. A selected element which is part of a hotlink will be marked with hollow squares. If we select several hotlinks, their selection dots will have different colors to help us distinguish them.



Part of HLM (Room 1.mod):
Object (Selected)
Name: Sofa Layout 21
Base Elevation: 0,00
Layer: Interior - Furniture
Story: 0. Ground Floor
Graphic Overrides:
pens black

Image 2. - Hotlink selection

A hotlink's content is part of the host project and we can select the hotlink elements and apply view options and overrides to them, but since these elements are locked in the host project we cannot change the element settings such as building material, height, etc. Even when a hotlink's source location is unavailable from the host, the hotlink content is visible in the host.

When we are **placing a hotlink**, in addition to mirroring, rotation and offsetting from a story level, we can adjust the hotlink instance either to the host's or its own story structure.

Read Help Center article [2] Place Hotlink.

The **Master ID** that we can assign to a hotlink allows us to schedule or locate it using the 'Find & Select' command. We can assign different Master IDs to the hotlinks with the same source.

The **Master Layer** of a hotlink controls the visibility of the entire hotlink instance. Furthermore, elements belonging to a hotlink remain on their own layers. If the hotlink's Master Layer is visible, we can still control the visibility of the elements by their own layers.

Read Help Center article [3] Hotlink Settings.

Let's see the next level of a hotlink – the **nested hotlink**. Simply put, it is a hotlink inside of a hotlink. It adds design flexibility, although it can represent the risk of oversights in the design project.

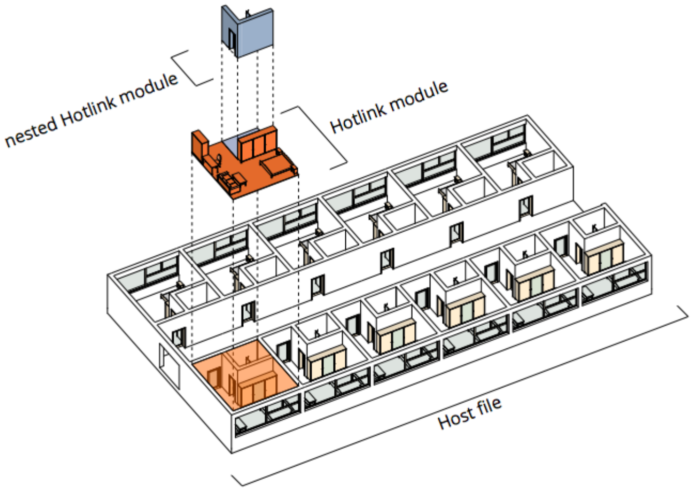


Image 3. - Nested hotlink concept

This issue is discussed later under the Hotlinks in detail section.

According to the type of the linked file, we can distinguish data formats that we can use as hotlinks in ARCHICAD:

- Hotlinks of ARCHICAD files (module, solo or Teamwork project files),
- Hotlinks of IFC files,
- Hotlinks of Rhino files,
- Hotlinks of Revit files[1].

The non-native ARCHICAD formats used as hotlinks can only reference geometries.

[1] With RFA & RVT Geometry Exchange Add-on. Read more: http://www.bim6x.com/solutions

Module (**.mod**) is an ARCHICAD format specially created for the purpose of hotlinking, this is the one we use most often.

Although saving a module file is possible from the File menu, we usually use the batch method of **Publisher Sets** to publish .mod files. The process of creating a dedicated Publisher Set is a very useful way to organize parts of the model and saving them in one step. Within a Publisher Set we filter the story views which will become hotlinks.

Saving .mod files from the Publisher also helps to determine **single** or **multistory** hotlinks more easily in the saving options of the Publisher Set's format. A single story hotlink will only filter the elements from the selected story. A multistory hotlink contains elements from a range of mutlitple stories. Single story hotlinks are more often used in practice because most elements are single story by nature.

For the hotlinking techniques introduced in this document it is important to point out the two most commonly used ways to place a **single story hotlink**:

- Publishing – We publish a .mod file from a file onto an external location. Then we place it as a hotlink into another or back to the same file.
- Linking – A hotlink has a reference to a particular single story of a .pln or Teamwork project. The source of a hotlink reference can also be a host, we can hotlink a story to the same file.

Module files store only the **attributes** that are used by their own elements. If there are differences between attributes, library parts or properties between a host and a hotlink, there are certain rules on how they are handled when we are placing a hotlink. Different attributes, libraries and properties could potentially cause many discrepancies in the files. There can be conflicts with differently named attributes that contain the same parameters or attributes with the same name but different parameters in the host and hotlink. To help avoid them we discuss the attribute management of hotlinks in a separate chapter: Hotlinks in details.

1.2 Hotlinks in practice

Users often identify elements as hotlink material if they are part of a functional group or repeat regularly. The use of a hotlink should rationalize or improve some aspects of the project workflow and not complicate it.

Here are some typical hotlink scenarios:

- representing repeatable parts of the model,
- documenting design options,
- isolating a functional part of the model,
- providing a reference to data coming from other authoring tools,
- enclosing construction elements of a project (a building core, a façade),
- representing entire buildings in a site project/urban study,
- restricting access to elements of the project.

Repeatable parts are the most well understood hotlink method, though new reasons for using hotlinks appear in large projects due to the need for more sophisticated requirements.

For instance, a surgery room represents a package of information that is determined by a board of hospital managers and doctors. This information would be best attached to a project as a hotlink, since we might want to store the notes of the meetings that led to the final design decision in the same folder. We may also wish to restrict access or limit unwanted changes to this particular part of the overall model.

Models originating from different disciplines (IFC format) are often placed directly into ARCHICAD projects in the form of hotlinks since the source data changes in complete blocks. Project stakeholder data placed as hotlinks allow for easy updating in the host project. Since a hotlink is not modifiable directly in the project, we can restrict the changes that are made to it to some degree by determining who has the access to the source. Furthermore, we can assign individuals to work on specific hotlinks.

These examples are brought up to highlight the value of hotlinks, but the list can be extended.

2. Hotlink techniques

Juggling the variables of project size, complexity and project use, as well as the composition of the team working on a project is hardly a simple task for even experienced hotlink users. The method of making the decisions less challenging involves relying on certain common hotlinking techniques:

1. The Simple method,
2. The Inventory file method,
3. The Iceberg file method.

2.1 The Simple method

An ARCHICAD model is split and then those parts are hotlinked into the host file.

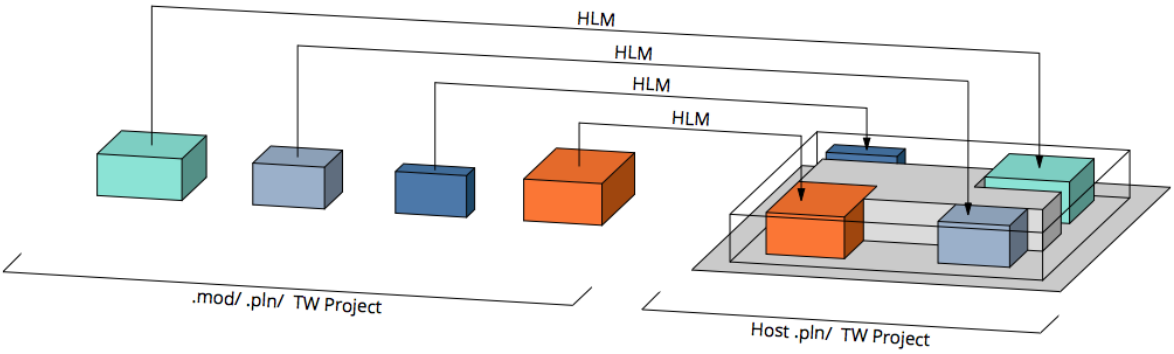


Image 4. - Simple hotlinking scheme

These parts are typically complete in themselves, such as a façade, a building wing or even an urban block.

Advantages

This is the most simple hotlinking process. It is used diversely, though it is recommended when the number of modules is not high.

Disadvantages

Attribute management can be tricky.

Managing the naming and updates of the hotlinks is challenging if there is a large number of hotlinks. It is a good idea to use predefined templates to avoid multiple, conflicting attributes.

To address these challenges, see the Hotlinks in detail section.

2.2 The Inventory file method

An Inventory file is used to systemize parts of a larger structure. It uses ARCHICAD's story system as an organizational filter and its publisher sets to simplify hotlink creation. Typically, an inventory file can have many stories, therefore it is important that we set up a structure to name the floors according to our design intentions.

We place the content of one module on one story. Sometimes we leave one or two empty stories in between a new inventory entry for design variants. Though new designs are frequently just added to the top of the "building" and we maintain a separate spreadsheet or document to reflect which floor a particular element is to be found on in the inventory file.

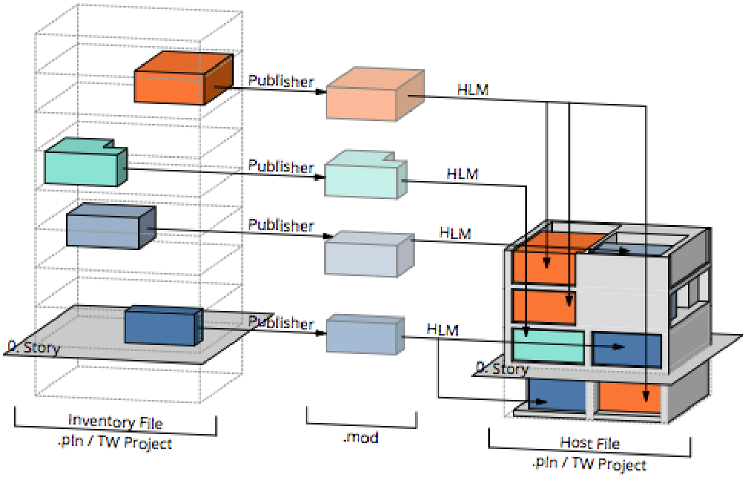


Image 5. - The Inventory file technique - Publishing modules

If we insert floors in an inventory file which uses automatic naming for publishing modules (automatic IDs for the story views), it might have the undesired effect of automatically renaming the published hotlink files whose content resides on floors above the insertion. For instance, a hotlink file as it is found in a file system might be called '15_storyBathroom.mod'. This is the name used when it is hotlinked into a host file. If we insert a new floor underneath it, then its name would change to '16_storyBathroom.mod' and the path of the hotlink in the host would become invalid.

This workflow is a favourite for hospitals, but residential or office towers can benefit from this method especially when the knowledge is reused in multiple projects.

In the illustration below one more step is added to the workflow, a fit-out file. This way the inventory file technique becomes even more powerful. We assemble parts into more complex modules that we place into the host file. We can make several groups from the same set of parts, making it easy to develop design options.

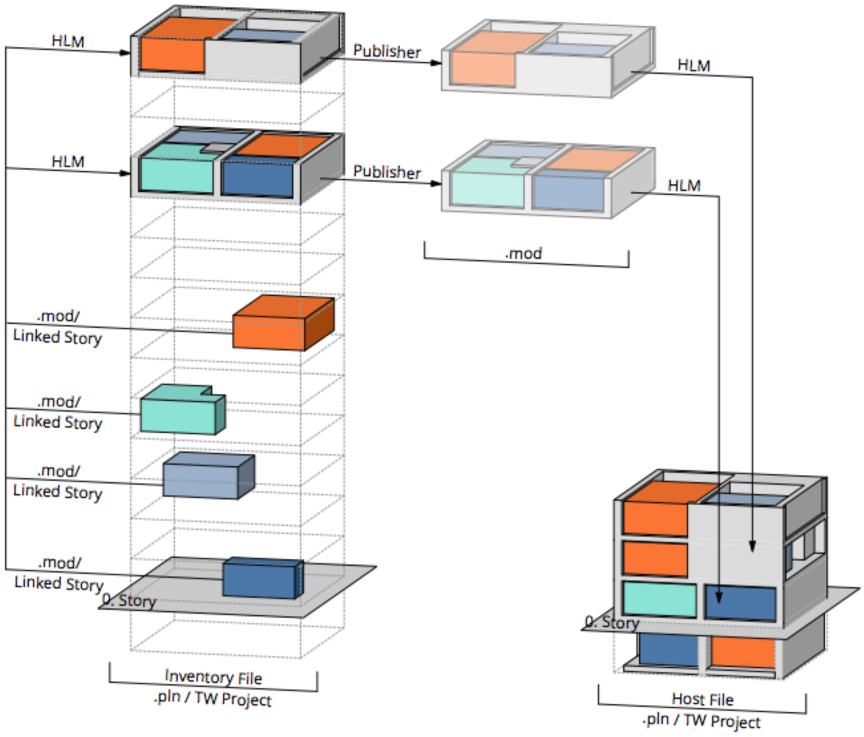


Image 6. - The Inventory file technique - Publishing Floor layouts

Advantages

- Attribute management is centralized in this one inventory file.
- Knowledge that is the basis of a complex project is also centralized.
- Same inventory file can be used in multiple projects.
- Typically used for hospitals.

Disadvantages

The upfront planning of a story structure is necessary, since inserting new stories into the existing structure of the inventory file changes the name of the output hotlink. This would require that we relink the hotlinks in the host file and would complicate already existing publisher set paths.

A range of stories normally addresses a certain design goal (apartment fit-out, bathroom design, shower enclosure types, etc.) which means we might need some empty floors for unexpected future changes in the project, or as mentioned we should document the story structure in a spreadsheet.

2.3 The Iceberg method

The model is just the tip of the iceberg. The sources for the hotlinks are literally under the surface in negative stories. It is not unusual to have hundreds of negative stories when we use this technique.
Virtual trace is important here, since unlike the inventory file and the simple technique, we don't have to design in a vacuum, but we can use 'trace reference' function to display columns, building boundaries, elevator shafts/core, essentially all of our model elements.

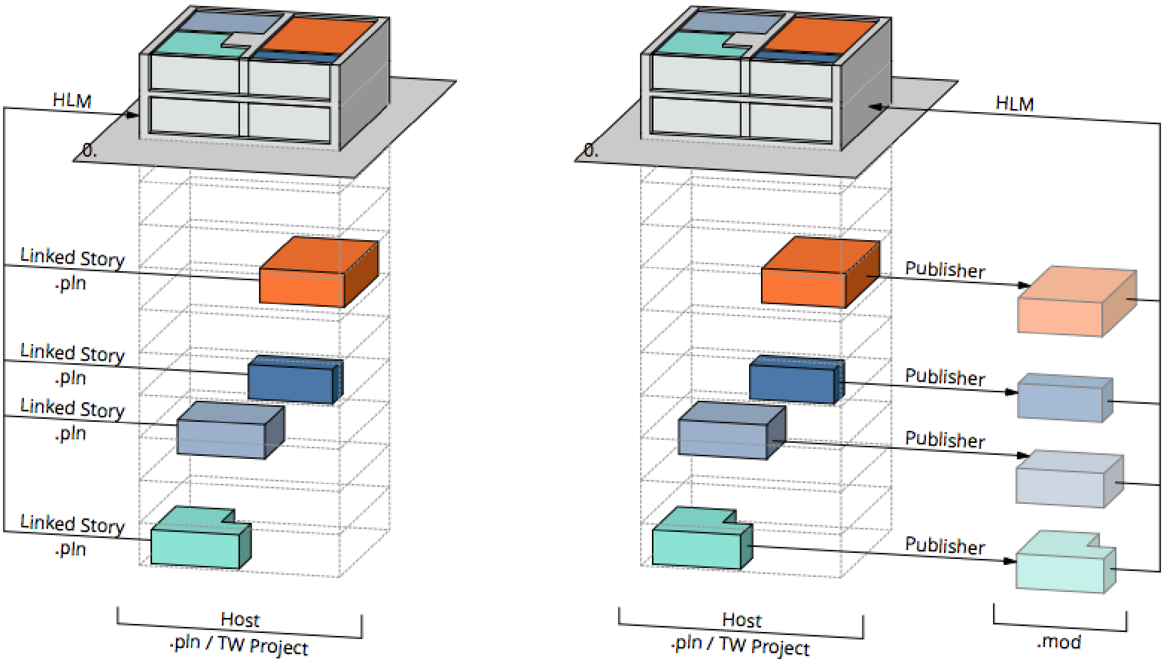


Image 7. - Simple Iceberg – a) Linked Story method, b) Publishing method

Advantages

- Easy to use with 'trace reference'.
- Management of both attributes and hotlinks are in one file.
- Typically used for high-rise buildings.

Disadvantages

It is difficult to develop a logical and consistent negative story structure, for the same reasons we mentioned in the Inventory file technique.
For the required views, lists and schedules we must filter the appropriate range of stories.

As in Image 8, we may group a range of negative floors into meaningful units. For instance, fit-out versions for a particular floor, or functional elements used in multiple stories.

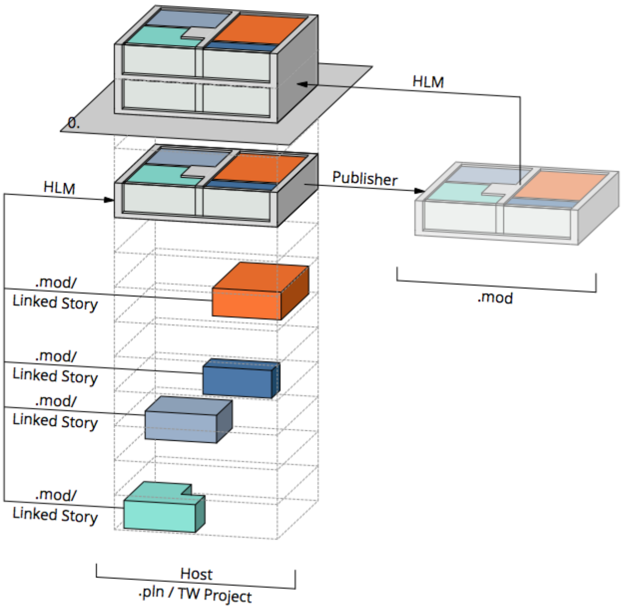


Image 8. - Simple Iceberg - Composing Floor layouts

The hotlinks are often linked to another negative story just to compose a more complex unit. Even the complex units can have multiple versions, see Image 9.

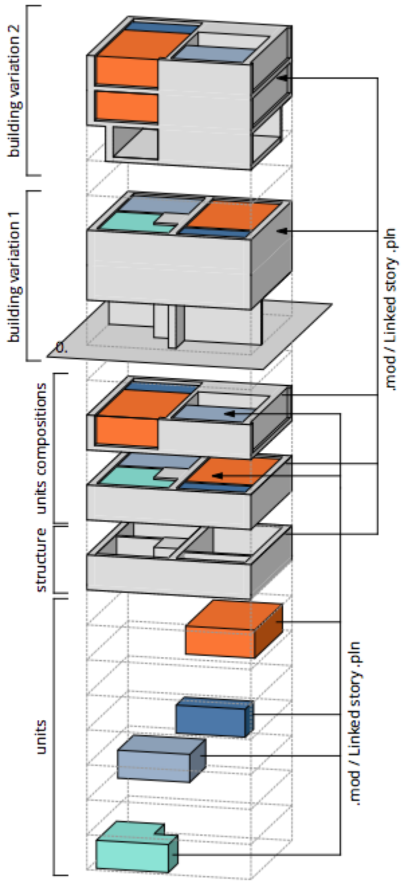


Image 9. - Versioning Iceberg method

3. Hotlinks in detail

Choosing the appropriate hotlink technique for the project is not the only challenge we have to address. There are several decisions we have to consider in connection to hotlink management.

If we look at the image of the Iceberg technique below, these questions arise: Should the box that is the hotlink be bound with a wall or should that wall be in the host? What type of data can be made into a hotlink? Should the hotlink be single or multi-story? Should we make more options with nested hotlinks? Where do the attributes that make up profiles come from? The host or the hotlink or a mix? What about the objects? Does the hotlink have its own library, or can one be embedded?

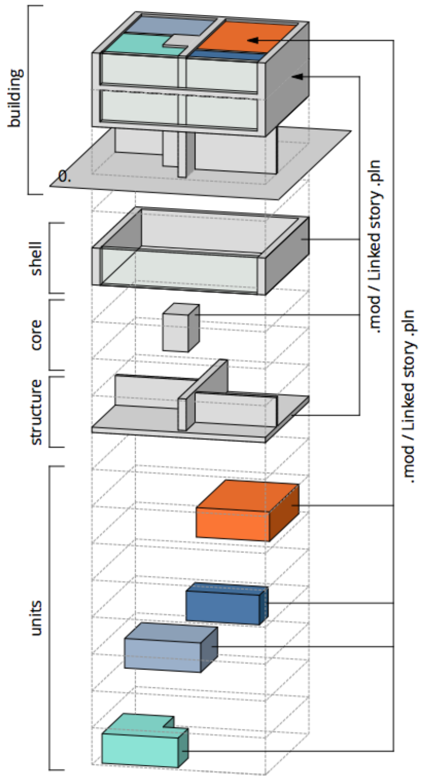


Image 10. – Iceberg technique

3.1 Common elements between host and hotlink

'The common wall' problem often relies on whether the wall is structural/load-bearing or not. Structural elements often remain part of a host project or can in their entirety become a separate hotlink. We separate structural elements from normal hotlink content most often in the Iceberg method, where we use Trace Reference to display structural elements in stories below where the interior units are composed. If the hotlink is a collection of objects arranged in a special way, e.g. a handicapped bathroom, we can place hotspots in the hotlink to show where we want the walls to be.

3.2 File types of hotlinks

The source file of a hotlink can be:

- a Module-type file (.mod)
- a Solo Project (.pln)
- an IFC file (.ifc)
- a Teamwork Project from a BIM Server/BIMcloud (of the same ARCHICAD version as a host project)
- a Rhino file (.3dm)
- a Revit file (.rvt)[2]

Most commonly we use the **.mod format** because it is optimized for hotlinks. This is the smallest in file size of the above-mentioned formats and we can publish it from the Publisher, most often just a single story. Creating modules through Publisher controls better saving options, locations and updating.

The basic properties of a **.mod** file are:

- It does not contain Section/Elevation/IE/Worksheet/Detail/3D Document or Layout data.
- It does not save library link information. Non-embedded library objects are inherited from the host file.
- It does not save unused attributes.

The best practice with a .mod file is to publish it and not to directly edit it. A .mod file is not meant to be used as a working file but more as a courier of information.

The reason for this practice is that the attribute and library information used in the .mod file is inherited from the most recently opened ARCHICAD project. If an earlier opened ARCHICAD project referenced different attributes or libraries than those we used in the host, we could unintentionally introduce those attributes/library objects into the .mod file which is open for editing. After we update the hotlink in the host, the hotlink will introduce a new attribute to the host project or show missing library objects.

Read Help Center article [4] Create Module (.mod) File.

Whereas if the .mod file is published from ARCHICAD, the following types of data are often directly edited in ARCHICAD and we would likely want them to be in **.pln format**:

- A site,
- A grid system,
- An attribute set,
- A Façade.

Essentially, any large data group, template or intellectual property of other project stakeholders that we might need to edit in ARCHICAD is a candidate for the .pln format.

[2] With RFA & RVT Geometry Exchange Add-on. Read more: http://www.bim6x.com/solutions

We advise setting up and checking translator settings in an empty ARCHICAD file before hotlinking an **IFC** file into the host project. To avoid introducing unwanted attributes or duplicates, this empty file has to have the same attributes and settings as the host.

3.3 Multi-story hotlinks

We use a multistory module when it contains an object that is truly multistory such as a stairway, an elevator shaft or a façade. When we save or publish a multistory module file we can specify a range of stories. Elements whose home story is out of the hotlink's story range may be included in the hotlink, if they are visible in the range of selected stories. When placing a multistory hotlink, we can select a specific story to be imported. If we want to bring in the whole range of saved stories, we need to place the hotlink again.
We have to pay attention to the possible story setting differences between a host and a multistory hotlink.

Read Help Center article [5] Place Module from Multistory File.

3.4 Nested hotlinks

Nested hotlink data is embedded in the host project. Similar to normal hotlinks, a nested hotlink will still be visible even if its source file is inaccessible.
When we skip a nested hotlink during the placement of a hotlink, the data of the nested hotlink is still in the host project but it is not displayed. In addition, we can control the display later in the Hotlink Manager.
The nested hotlink can be considered a design option to a hotlink. Thinking of it in this manner is a double-edged sword. Small changes in the nested hotlink can have large consequences in the host project. We might overlook the significance of a nested hotlink and we could accidentally make unwanted global changes in the host project. Therefore, it is common to choose the '**break nested hotlink**' option during publishing. This way when we place a published hotlink into a host, it represents a fixed design choice and cannot be re-evaluated unintentionally.

3.5 Managing hotlinks

Any hotlink placed in the host project will be listed in the **Hotlink Manager**. ARCHICAD's Hotlink Manager dialog provides an overview of the hotlink's hierarchy and a feedback on the status of the individual source files (up-to-date, missing, modified) and their source location. We can update the entire hotlink structure in one step, or just an individual source file at a time. We can most easily manipulate the operations of saving, deleting, breaking, relinking, changing hotlinks through the Hotlink Manager. Updating, saving, breaking and opening to edit in a separate ARCHICAD file is available from the drop-down context menu of the selected hotlink.

Read Help Center article [6] Hotlink Module Manager.

3.6 Attribute management with hotlinks

There is an ARCHICAD synchronization process of attributes when a hotlink is placed in a host. The rules of this process are the following:

- If an element of a hotlink uses an attribute with the same name as an attribute in the host project, the host attribute will be used.
- If no attributes of that name exist in the host, the new attribute of the hotlink will be added to the host project as the last attribute in the host's Attribute Manager list.
- If the attribute is part of the host file, that attribute will not be updated by any attribute modifications in the hotlink.
- If the attribute is modified in the host, the modification will not be reflected in the hotlink.
- If an attribute was imported by a hotlink placement and later that hotlink was deleted, the added attribute will not be removed from the host.
- If an attribute in a hotlink is a layer combination or a pen set, it does not participate in the attribute synchronization.

The Inventory File and Iceberg techniques we mentioned earlier define well how to centralize the management of the element attributes among multiple files.
Translator or import settings determine the synchronization of attributes that come from translated hotlink files (currently IFC, Rhino, Revit).

For large projects, or for projects with a large number of hotlinks, some users employ an empty file, frequently called 'attribute.pln', solely for the purpose of attribute management. In that file, we can find all the new attributes, usually created by a BIM Manager. If we need to make any changes, we have to execute it in this **central attribute management file** and with the help of the Attribute Manager we can overwrite the attributes in all the other files. This file does not hold any model data. Moreover, we can hotlink this file to the host and any other file used with the project to automatically bring over the attributes. In this case, users create samples of each element and attribute, placing and systemizing them in a dedicated "dummy" story. This is often included as the part of the company's template documents and files. This sample attribute library can be handy to pick up parameters for creating new elements as well. We must exclude sample elements on dummy stories from Schedules listing and IFC export, since they are placed in the model for internal management workflow.

Read Help Center article [7] Attribute Management of Hotlinks.

3.7 Embedded libraries in hotlinks

After we place the hotlink, the hotlink's embedded library will be added to the host's embedded library. A dedicated hotlink folder will be created in the host's embedded library. If the hotlink contains any library parts from a linked library, the linked library objects will not be imported into the host. In this case, we need to link the same library or add the hotlink's library into the host project.

Read Help Center article [8] Edit Hotlink Module Content.

3.8 Element properties in hotlinks

If a hotlink contains new properties that are not contained in the host project, they will be shown greyed out in a Property Manager of the host. We can search and schedule them, but in order to edit or delete a property from the host we need to merge it into the host from the Property Manager dialog.

Read Help Center article [9] Element Properties in Hotlinks.

4. Special Considerations

4.1 Project performance and hotlinks

As a teamwork project develops and gets more complex, we might experience performance issues. The typical issues might be:

- join time becomes longer and “post processing” is the longest phase,
- certain operations tend to evoke the wait cursor for long periods of time,
- the use of the Hotlink Manager becomes logistically challenging.

As a project grows it will inevitably slow down, therefore the changes in the workflow will not be enough to achieve the desired performance. However, we can often avoid the slowness.

Typically, the hotlink loses its design advantage in the final stages of design development. Let us name examples: In residential buildings with repeated apartments in the final stages of the project design customers have special requirements, like removing dividing walls or reorganising a bathroom; the structural column size changes across the building; labelling should be different. In those cases, it is useful to break a hotlink and make custom changes to the instance in question. All in all, when there are no longer changes in the hotlink, it is best to break it, since the number of hotlinks increases the post processing time of the join and affects some update operations.

One of the complications that can arise when using the Inventory File or Iceberg techniques is the increased number of unnecessary embedded libraries. It all starts with just one embedded library in the project. When we publish a hotlink from this project, the hotlink will automatically have the main file’s embedded library embedded into it. To overcome this, we should avoid having an embedded library in the project from which hotlinks are published.

Next part of this growing issue is when we break a hotlink that has an embedded library, the hotlink’s embedded library will be embedded into the host. This process increases the number of embedded libraries and will degrade the performance of the host file.

4.2 Hotlinks in Teamwork projects

A hotlink which has a source from a Teamwork Project will often misleadingly appear out of date. The portion of the project which was hotlinked in may have remained static, but the sent changes that affected other parts of the project make the hotlink in the host appear out of date. With a .mod file hotlink this update status is more fine and we can control it more precisely. For this reason the process of updating and keeping up to date Teamwork hotlinks is a bit more challenging than keeping a .mod hotlink up to date. In the case of the .mod file the Hotlink manager has less operations to check to determine whether something should be updated. The appearance of 'out of date' hotlinks which source from a Teamwork project may be caused by unsent changes in Teamwork, even when the model is up-to-date.

The 'break' command is not available for hotlinks in a Teamwork project. However, we can explode individual hotlink instances. If we have any dimensions or Solid Element Operations connections in the hotlink, they will be lost.

After reserving all instances, the Hotlink Manager must close for data structure safety reasons and ARCHICAD does this automatically. In general, it will re-open and we can continue to work in the dialog. Sometimes we will need to manually re-open the Hotlink Manager dialog.

If there is an ARCHICAD version difference between the host Teamwork project and the hotlinks in .pln or .mod file format, there could be a significant delay in the join process; because there is a version upgrading of hotlinks during the join process in the background. This delay will repeat with every join process until we upgrade the hotlink to the same ARCHICAD version as the host Teamwork project. Hotlink in a Teamwork project will emit a warning message about this.

Before hotlinking an IFC file into a Teamwork host project, make sure we have all the setting permissions. Otherwise, those settings and attributes won't be imported even though they are set up in the translator.

Read Help Center article [10] Hotlink Modules in Teamwork.

5. Best practices

- Hotlink workflow should be documented in a BIM manual for the company and it should include a scheme of hotlink hierarchy in projects, nomenclature and also such topics as location of hotlinks, responsible managers and accepted practice.
- The BIM Manager should be attentive to attribute management. That is why it is important to have a detailed template file right from the beginning that contains most of the attributes that will be needed throughout the lifecycle of the project.
- Hotlinking a single story .mod file in general offers better performance than the same data in multistory hotlink format or .pln format.
- Avoid using a .pln file where we can use a .mod file. Updating time is faster and the size of the file is smaller.
- In a case of using a hotlink as a design option, the hotlink's Master Layer should have a different priority number in order to avoid unwanted building material collisions.
- In a Teamwork project reserve the hotlinks we want to relink before we begin relinking them in the Hotlink Manager.
- If there is a need to change the source of many hotlinks in a Teamwork project, it is easier to do it in a saved out .pln file and then re-sharing it.
- In case of small variations at the final stages of the project: break the hotlinks! Avoid increasing their number.

Appendix

Help Center links

[1] About Hotlink Modules https://helpcenter.graphisoft.com/?p=65658
[2] Place Hotlink https://helpcenter.graphisoft.com/?p=65659
[3] Hotlink Settings https://helpcenter.graphisoft.com/?p=65661
[4] Create Module (.mod) File https://helpcenter.graphisoft.com/?p=65663
[5] Place Module from Multistory File https://helpcenter.graphisoft.com/?p=55544
[6] Hotlink Module Manager https://helpcenter.graphisoft.com/?p=65662
[7] Attribute Management of Hotlinks https://helpcenter.graphisoft.com/?p=25589
[8] Edit Hotlink Module Content https://helpcenter.graphisoft.com/?p=65660
[9] Element Properties in Hotlinks https://helpcenter.graphisoft.com/?p=65299
[10] Hotlink Modules in Teamwork https://helpcenter.graphisoft.com/?p=65655

Keywords

Hotlink Module, Hotlink, Hotlinking, ARCHICAD