Skip to content

Library Overrides

Library Overrides is a system designed to allow editing linked data, while keeping it in sync with the original library data. Most types of linked data-blocks can be overridden, and the properties of these overrides can then be edited. When the library data changes, unmodified properties of the overridden one will be updated accordingly.

Note

The old proxy system has been deprecated in Blender 3.0, and fully removed in Blender 3.2. Automatic conversion from proxies to library overrides happens when loading a blend-file, but results on complex characters are not guaranteed and may need manual fixes.

Library overrides supports:

- Multiple independent overrides of a same linked data (e.g. having the same character multiple times in the same scene).
- Adding new modifiers and constraints, anywhere in the stack.
- Recursively chaining overrides (i.e. link and override overrides from another library file, etc.).

Note

There are known issues that have to be addressed. See the main task of the project, for more details.

Warning

While in most cases library overrides data is preserved across a loss of reference linked data (if e.g. the library file becomes unavailable or is relocated) there are some exceptions.

The main one is probably posed (but not animated) armature objects, when their Armature obdata itself is not overridden. The Pose bones of an armature object are fully linked to the bones of its Armature obdata, if the later goes missing, the pose bones are definitively lost.

Note

Proper Collections Layout Matters

For library overrides to work well, it is much better if **all** the collections needed by the character are children of the root (linked and instantiated) one, such that there is a clear hierarchy. Otherwise, some data may not be properly automatically overridden, and other operations may be less reliable.

Override Hierarchies

Hierarchy is a very important concept to understand when working with library overrides. In Blender, a real-life asset (a character, a prop, a set, etc.) is almost never made of a single data-block, but is rather a group of data-blocks with dependency relationships to each-other. E.g. a character will typicall have an armature object, several geometry objects, rig-controllers objects, the object data for all of these objects, materials, textures, etc.

These relationships can be represented as a tree, with a root data-block 'linking-in' all its dependencies, recursively. With library overrides, typically, the root of the hierarchy is also the data-block that is directly linked when importing the asset (usually a collection).

This concept of hierarchy can also be seen as some sort of super meta-data-block. It is critical when there are several overrides of the same linked data, since it allows to clearly identify a given data-block to one override, leaving no ambiguity to processes that affect the whole hierarchy (e.g. resyncing overrides with their linked data). It also allows to share relationships between data-blocks of different hierarchies, like a parenting relationships between two different overrides of a same character.

Animation & Overrides

Due to current design of animation data in Blender, what is editable in overrides' animations can change greatly depending on whether animation data wa already defined in the linked reference data-block. Animation data is created for a datablock if it gets animated by keyframes, or through drivers.

In general, an overrides can do much more with its animation data if no animation data exists in its linked reference data-block.

Keyframes (a.k.a. F-Curves)

Keyframed animation belongs to another data-block (an Action one). So it is possible to assign a purely local Action data-block replacing the one linked from the library. This will completely replace the keyframed animation from the linked data though, and not override it in any way.

Overridden Action data-blocks only support a very limited amount of editing. For example, an existing F-Curve can be muted, but its keyframes cannot edited, and no new F-Curve can be added.

Drivers

If the linked reference data has animation data, then its overrides only have limited possibilities to edit the existing drivers. For example, it will be possibl to change the exisitng target of a driver, but it won’t be possible to add new drivers, or new targets to an existing driver.

If the linked reference data has no animation data, then its overrides will create a new one when they get some drivers defined. Drivers can then be fully edited, added or removed, just as with purely local data-blocks.

NLA

The NLA editor data also belongs to the animation data of a data-block. However, this data does support some greater level of editing in overrides, including moving or resizing existing strips from the linked data, and adding new local strips.

Resyncing Overrides

The relationships between linked data-blocks can change, resulting in outdated overrides. When this happens, overrides need to be resynced to match th new structure of their hierarchy. Overrides are automatically resynced if needed on blend-files opening. However, it may be needed to resynced them manually sometimes, see Troubleshoot an Override Hierarchy.

Tip

Blender is also able to resync library overrides from external libraries, that are then linked into a working file. However, this is a costly process that needs to be fully redone every time the working file is loaded, since Blender cannot edit/modify the external library directly.

So users linking overrides (or creating recursive overrides) should ensure that their library files are regularly updated, to avoid this overhead on file load (typically, opening and saving those library files should be enough to update them).

Tip

Auto resyncing can be disabled in the Experimental Preferences.

Non-Editable Overrides

For technical reasons (how relationships between data-blocks are stored), Blender needs to create overrides of a lot of data-blocks, even when only on or two of them actually needs to be edited by the user. To reduce the amount of information and risk of potential unwanted editing, most of these data-blocks are now marked as non-editable by default. This can be changed once the override has been created.

Make an Override

Reference

Editor:
3D Viewport, Outliner, Properties

Mode:
Object Mode

Menu:
3D Viewport ‣ Header ‣ Object ‣ Library Override ‣ Make Outliner ‣ Context Menu ‣ Library Override ‣ Make ID Widget ‣ Context Menu ‣ Library Override ‣ Make

Shortcut:
`Shift - LMB` on the ‘linked’/’overridden’ button of an ID Widget.

Create overrides from the selected data-blocks.

Blender automatically create overrides for all required data-blocks to ensure that valid override hierarchies are created.

Only overrides created from selected items will be user-editable.

Warning

The support for the creation of library overrides from the ID Widget (mainly from within the *Properties* editor) is limited. While the most common usages should be supported, especially with Objects, meshes, etc., much remains to be implemented.

Selected Items

Depending on where from the override is created, there are several ways to 'select' items to be overridden and user-editable.

Note

This also applies to the other common operations (*Reset* and *Clear*).

The *Troubleshoot* advanced operations only available from the Outliner always apply to a whole override hierarchy.

3DView

The selected objects will be considered as selected.

When a selected object is a local *Empty* instantiating a linked collection, the following will happen: * The *Empty* object will be removed. * Its linked collection will be overridden, and that override will be instanced in the same collection in the current *View Layer*. * If the collection contains *Armature* objects, they will be user-editable. Otherwise, no created override will be defined as user-editable.

Outliner

The operation can be applied on either the selected items only, their content only, or both.

Tip

Using *Selected & Content* is an easy way to get all newly created overrides immediately user-editable.

ID Widget

Only the linked data-block in the ID Widget is considered as selected, and set as editable once overridden.

Make Editable

That same operation can also be used to make existing overrides user-editable, after they have been created, or cleared

Reset an Override

Reference

Editor:
3D Viewport, Outliner, Properties

Mode:
Object Mode

Menu:
3D Viewport ‣ Header ‣ Object ‣ Library Override ‣ Reset Outliner ‣ Context Menu ‣ Library Override ‣ Reset ID Widget ‣ Context Menu ‣ Library Override ‣ Reset

Reset the selected overrides to their original values (from the linked reference data). Unlike with the *Clear* operation, the overrides remain fully editable, and are never deleted.

Clear an Override

Reference

Editor:
3D Viewport, Outliner, Properties

Mode:

Mode:

Object Mode

Menu:

3D Viewport ‣ Header ‣ Object ‣ Library Override ‣ Clear Outliner ‣ Context Menu ‣ Library Override ‣ Clear ID Widget ‣ Context Menu ‣ Library Override ‣ Clear

Shortcut:

`Shift - LMB` on the ‘overridden’ button of an ID Widget.

Reset the selected overrides to their original values, and if possible without breaking the existing hierarchy, delete them and replace them by their linked reference data. Otherwise, keep the overrides but mark them as non-editable.

Edit an Override

Essentially, an override is edited the same way as a regular local data-block. You can use operators on them, edit their properties from various editors, etc. There are some limitations however, most notably Edit Mode is not allowed for overrides. In most cases, as soon as you edit a property, you can se that it’s overridden by its teal blue outline/background.

You can also animate overrides, animated properties just replace/supersede overrides then. Note that you cannot override/edit an existing animation, you have to create a new action. You can manually define or remove an override from the context menu of the relevant property. If an override is not editabl you have to make it editable first.

Define Overrides

Reference

Editor:

Any

Mode:

Object Mode

Property:

Context Menu ‣ Define Overrides Context Menu ‣ Define Override

Mark a property to be overridden in the local blend file. For array properties all elements will be overridden.

Define Single Override

Reference

Editor:

Any

Mode:

Object Mode

Property:

Context Menu ‣ Define Single Override

Mark a property to be overridden in the local blend file. For array properties only the selected element will be overridden.

Remove Overrides

Reference

Editor:

Any

Mode:

Object Mode

Property:

Context Menu ‣ Remove Overrides Context Menu ‣ Remove Override

Remove the property from the overrides. The value of the linked in data-block will be used. For array properties all elements will be removed from the

Remove the property from the overrides. The value of the linked in data-block will be used. For array properties all elements will be removed from the override.

Remove Single Override

Reference

Editor:
Any

Mode:
Object Mode

Property:
Context Menu ‣ Remove Single Override

Remove the property from the overrides. The value of the linked in data-block will be used. For array properties only the selected elements will be removed from the override.

Troubleshoot an Override Hierarchy

Reference

Editor:
Outliner

Mode:
Object Mode

Outliner:
Context Menu ‣ Library override ‣ Troubleshoot

These operations are only available from the Outliner contextual menu. They can help fixing a broken override hierarchy.

Resync

Reference

Editor:
Outliner

Mode:
Object Mode

Outliner:
Context Menu ‣ Library override ‣ Troubleshoot ‣ Resync

The hierarchy of the linked data (the relationships between linked data-blocks) can change. Overrides need to be resynced to match the new hierarchy. This operator will resync the override to match the new hierarchy in the library.

Warning

While resyncing a library override it is possible that edited overrides get deleted if they are changed in the original library. If this is the case, a warning message will be displayed stating how many overrides were deleted, if the deletion is undesirable the resync can be undone before saving the blend-file.

Note

This Process is Automatic

Usually, this operation happens automatically when blender detects it is needed, on file load, unless it is disabled in the Experimental Preferences.

Resync Enforce

Reference

Editor:

Editor:

Outliner

Mode:

Object Mode

Outliner:

Context Menu ‣ Library override ‣ Troubleshoot ‣ Resync Enforce

In some cases, especially with older blend-files that were saved with ‘broken’ (non-hierarchy-matching) overrides, a regular resync itself cannot rebuild properly the override as expected (e.g. some objects might go missing). To solve this issue, this operator rebuilds the local override from its linked reference, as well as its hierarchy of dependencies, enforcing that hierarchy to match the linked data (i.e. ignoring existing overrides on data-blocks properties). This is similar to a regular resync, but is more forceful, aggressive, at the cost of a potential loss of some overrides on ID pointers properties.

Delete

Reference

Editor:

Outliner

Mode:

Object Mode

Outliner:

Context Menu ‣ Library override ‣ Troubleshoot ‣ Delete

Remove the whole library override hierarchy, and replace all of these override data-blocks by their original linked data-blocks. This fully reverts the *Mak* operation.

Ne
Asset Librar

Previous
Link & Append

Made with Furo

Last updated on 2025-05-10

View Source
View Translation
Report issue on this page