

3D Digital Backbone Driver for Blender

About this document

This is the documentation to install and use the 3D Digital Backbone Driver add-on for Blender by LUMISCAPHE. It is up to date with version 1.0.0.

Downloads and ressources

This software and its resources are available at https://lumiscaphe.com/3ddbb_blender_driver.

License

The add-on source code is LGPL licensed by LUMISCAPHE (lumiscaphe.com).
The add-on embeds a DS Viper for Python runtime wheel component licensed with an Open License from DIGITAL SUBSTRATE (digitalsubstrate.io).

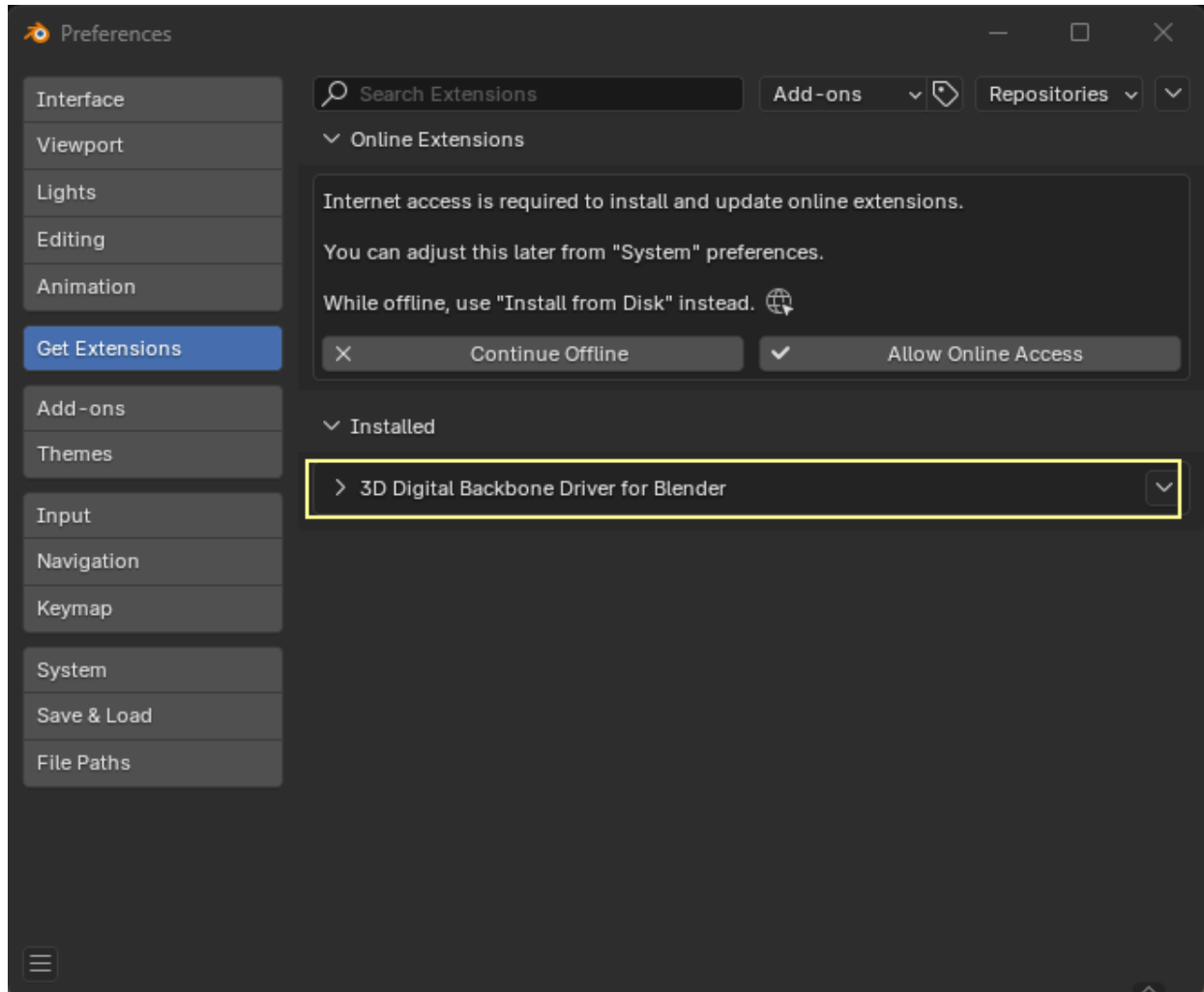
By using the add-on, you agree to comply with these licenses, the texts of which are available in the add-on folder once it has been installed.

Installation guide

Requirements : Blender 4.5 (version 4.3 and 4.4 are supported as well)

The extensions are delivered as a .zip file, which can be installed following these instructions <https://docs.blender.org/manual/en/4.3/editors/preferences/extensions.html> and choosing to install from disk the extension **3D Digital Backbone Driver for Blender**.

Once installed, the extension should appear in the list of installed extensions, and activated as addons in the Edit / Preferences menu / Add-ons tab.



[extension_installed.png]

Why to use the 3D Digital Backbone

The connection to the 3D Digital Backbone give some advantages :

- **Historization of the data.** Each save is recorded in the 3D Digital Backbone. And it is possible to restore any state of the data, knowing its “commitId”. This feature needs additional development to let find available commitIds
- **Collaboration.** Synchronize data between multiple instances of Blender, and collaborate working together on the same model.
- **Interop.** Convert Blender data to other formats using 3D Digital Backbone converters,

Vocabulary/Context

In this documentation the following concepts will be used several times :

- **.blend file / Blender data file:** the native file format used by Blender to save/load data.

- **3D Digital Backbone, simplified as DBB:** database used to store the Blender database which corresponds to the content of a .blend file. A local database file is used in this documentation. The file extension can be anything, in this documentation .dbb is used. A DBB can contain multiple Blender databases, meaning that in Blender, when loading content from a DBB you have to first select the filepath and then select the name of the Blender database to load, it is a two step process.
- **Database name:** a name associated with the Blender database saved in the DBB, as multiple Blender databases can be saved in one DBB, the name acts as the identifier for the Blender database. This name is chosen by the user.

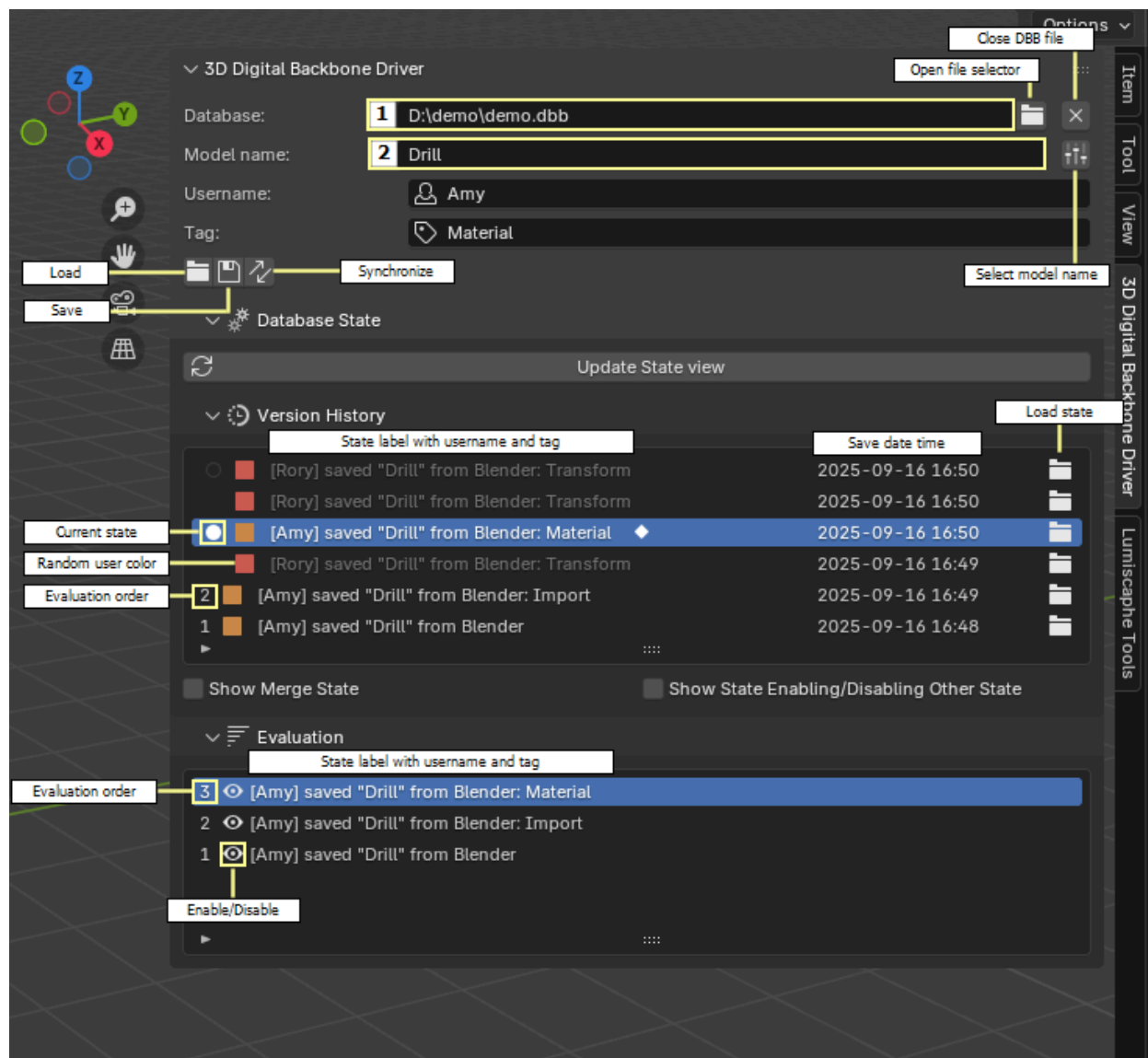
Supported data

The following properties in each data-blocks are saved :



- Scene: name / collection children / objects children
- Collections: name / colors / hide_status / collection children / objects children
- Objects: name / data binding / transform / parent / hide_status
- Meshs: name / mesh essential data (vertices/edges/loops/polygons)
- Materials: only nodal materials and the nodes Output and BSDFPrinciple
- Light: name and for Point Light the power property

For cameras, no specific properties are saved (only its properties as objects are saved).

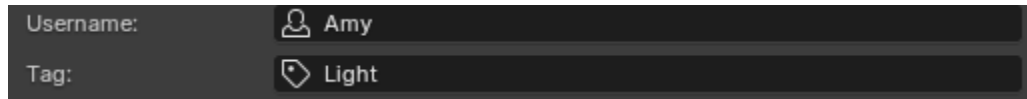
User interface



How to select the Blender data in the DBB

1. Click on the file selector in the first row  to choose the 3D Digital Backbone database file to load : select an existing file in which the content will be saved. If the file does not exist, the file will be created at the first save operation
2. Choose a Blender database name :
 - if it is the first save, write a name in the Name field
 - If it is a model already saved in the DBB, click on  to display the names of the models stored in the DBB

Customizing the version label

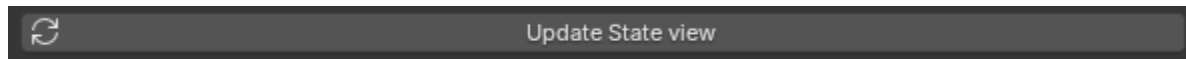


Each time the save button is pressed, the current state of the data in Blender will be saved in the DBB along with a label composed of the username and tags. In the interface, by default the username is set to the user name of the computer. If the username field is left empty, in the version label it will be mentioned "Anonymous user".

The tag which is optional can be helpful to identify the version.







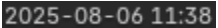

Version History

The menu Version History displays the list of states saved in the chosen DBB. To update this list, press the button




The list is not automatically updated.

In the list, there is

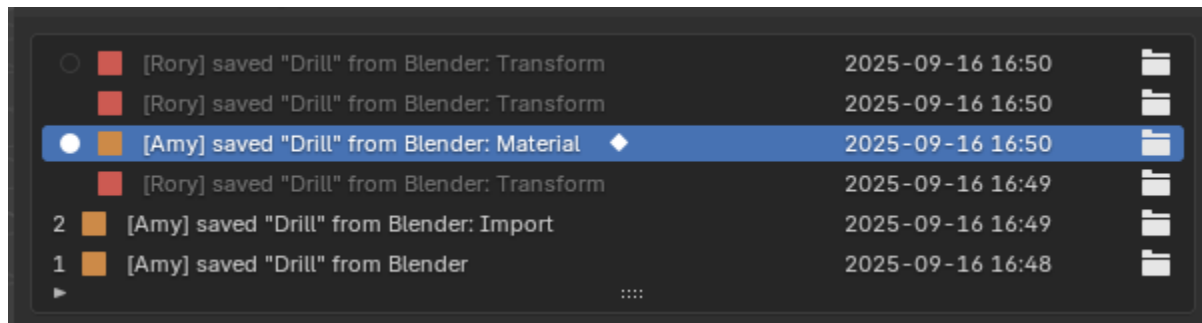
- The state type which can be :
 -  The current state
 -  The most advanced state in another branch
 -  the state with a number means that this state is used in the evaluation of the current state and the number indicates the evaluation order
-  A color attributed randomly to each user
-  The label composed of the username, the data name, the application from which the data was saved and the tag
-  Eventually this icon, which indicated that it is the current version from which the data was loaded in this Blender session
-  The date when the save occurred
-  Button to load the corresponding version

By default, merging state and enabling/disabling state are hidden. To display those click on

these checkboxes  Show Merge State

 Show State Enabling/Disabling Other State

An example of a state list :




The list can be read as :

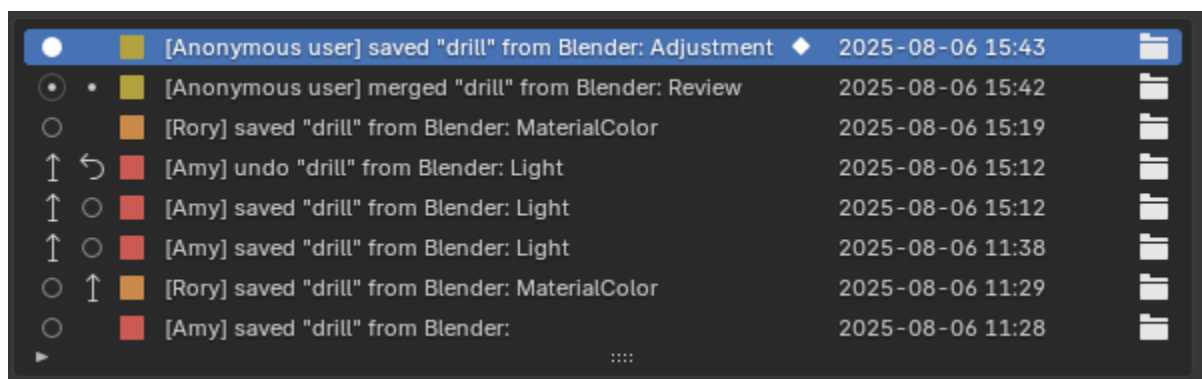
1. User [Amy] initialized the data "drill" from Blender
2. User [Amy] saved the data "drill" from Blender with the tag [Import]
3. Starting from the state 2, user [Rory] saved the data "drill" from Blender with the tag [Transform]
4. User [Amy] saved the data "drill" from Blender with the tag [Maerial], this is the current state
5. User [Rory] continued his work by saving twice later.

Here the current state takes into account only the first two states and the current state.

Evaluation

This list allows the user to understand for a loaded state, ther order in which states were evaluated to compute the current state. By toggling the eye icon  the user can enable or disable a state, and the result will be automatically updated in the 3D Digital Backbone and in the opened Blender session.

For example, for this version tree the first user Amy first initialized the DBB. Then the second user Rory loaded this DBB and started working on the Material Color. In parallel, the first user Amy continued to work on her side, saved a few times but did an undo to remove the previous modifications. At last, an anonymous user merged the work from both users and started working in this DBB.



The versions are, in the chronological order :


6. User [Amy] saved the data “drill” from Blender
7. Starting from the version 1, user [Rory] saved the data “drill” from Blender with the tag [MaterialColor]
8. A new branch is starting from the version 1, user [Amy] saved the data “drill” from Blender with the tag [Light]
9. Starting from the version 3, user [Amy] saved the data “drill” from Blender with the tag [Light]
10. User [Amy] undo the version 4
11. Starting from the version 2, user [Rory] saved the data “drill” from Blender with the tag [MaterialColor]
12. Both versions from 5 and 6 are merged into one version
13. Starting from the merged version 7, an anonymous user saved “drill” from Blender with the tag [Mesh]

Operations

For databases containing a lot of meshes, the save operation and initial load operations can take some time.

How to initialize the Blender data representation in DBB from a .blend file

Context: The data is not in the DBB, and you want to save it.


1. Load a .blend file
2. [Select a DBB database path and model name](#)
3. [Optional] Set a user name and tag keyword
4. Click on the save button 

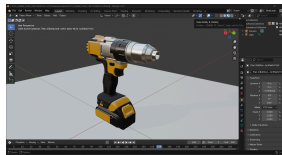


How to save the new modifications to the DBB

Context: The data is already saved to the DBB, and you want to save new modifications.

1. [Select a DBB database path and model name](#)

2. [Optional] Set a user name and tag keyword
3. Click on the save button 




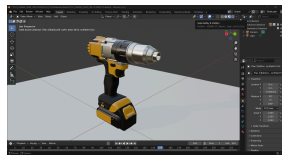
→ DBB

How to load Blender database from the DBB

WARNING: loading a DBB will clear all data from the current Blender session. It is safer to start from an empty Blender session.

Context: The data is already saved to the DBB, and you want to load in your current Blender session.

1. Open Blender without loading a specific .blend file
2. [Select a DBB database path and model name](#)
3. Click on the load button 
4. Your Blender is ready to be used !



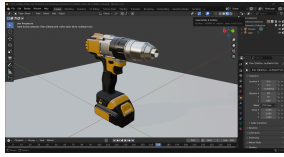
← DBB

How to synchronize the local Blender data with the Blender data in the DBB

Prerequisites : [Blender database is loaded from the DBB](#)

Context: The data is already saved to the DBB, you want to save all your modifications and merge them with all the recent modifications done by other users.

Choosing to click on the Synchronize button  means that your work will be saved, then all the modifications on the database will be merged and the result of it will be displayed in your current Blender session.




← DBB

How to close a DBB file

Prerequisites : [Blender database is loaded from the DBB](#)

Context: This functionality might be needed in the situation where you want to delete the DBB file but your current Blender session has the database opened: if you click on close, you can close the database without closing Blender.

Click on the close button . The DBB file has to be opened previously otherwise the button is disabled.