

User Data

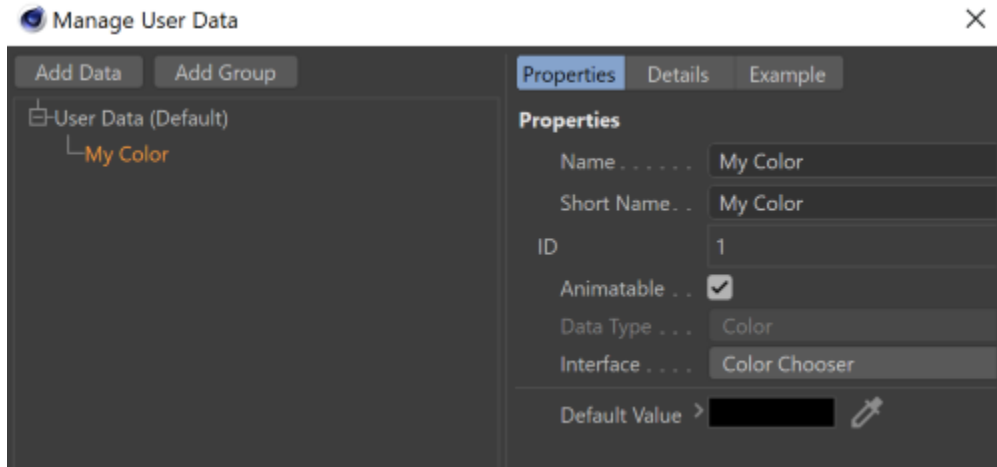
It is possible to create arbitrary user data that can be attached to any object. This user data can then later be used at shading time by specific nodes with a user prefix.



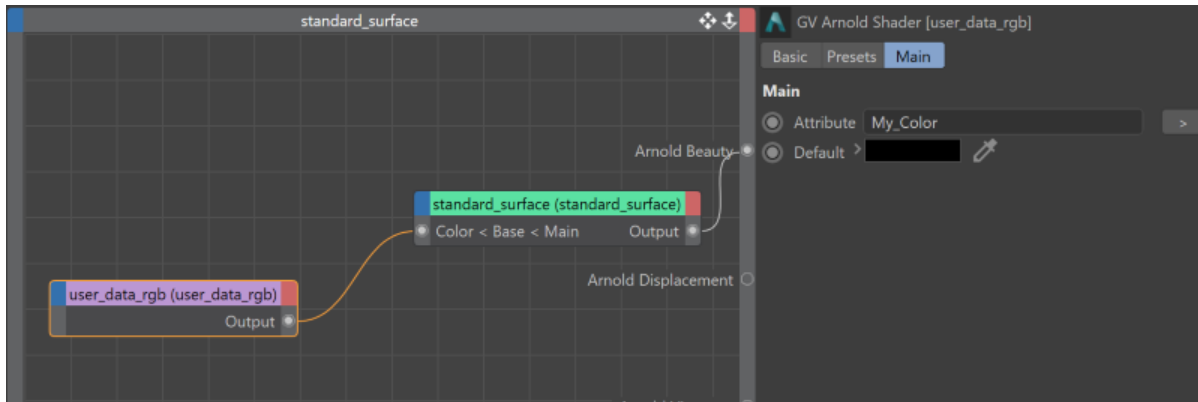
A video tutorial that demonstrates the use of the `user_data_string` shader can be found [here](#). A tutorial on how to use the `user_data_RGB` shader can be found [here](#). Information about User Data shaders can be found [here](#).

The convention used to store the user data can be found in different data types such as color, float, etc. For example:

- Create a Color type user data in C4D and call it 'My Color':



- Create an Arnold Material and read the 'My Color' user data via a `user_data_rgb` shader.





Only alphanumerical characters, dashes (-) and underscores (_) are allowed in the parameter name. Arnold converts all other characters (including spaces) to underscores. For instance in the above example 'My Color' becomes 'My_Color', which you have to set in the user_data shader. Also note that parameter names are case sensitive.

You can check the Arnold parameter name if you export the scene to an .ass file and look for the user parameter as follows:

```
en!-uan!-uyen!-u:el u7???? en!-u:el u7????/en!-uan!-uyy:en!-u????/yen!-u
nlist 24 1 b05VECTOR
zzyzzyzzyzzyz/iT0yzM/iT0yzM/iT0yzM/iT0zzaRT=dzzaRT=dzzaRT=dzzaRT=daRT=dz
uvlist 4 1 VECTOR2
0 0 0 1 1 0 1 1
smoothing on
disp_map NULL
step_size 0
declare My Color constant RGB
My_Color 0.839681984 0.63478744 1
}

gaussian_filter
{
  name default_filter
  width 2
}
```



If the display color of an object is enabled we will export the 'display_color' RGB as a user data.