

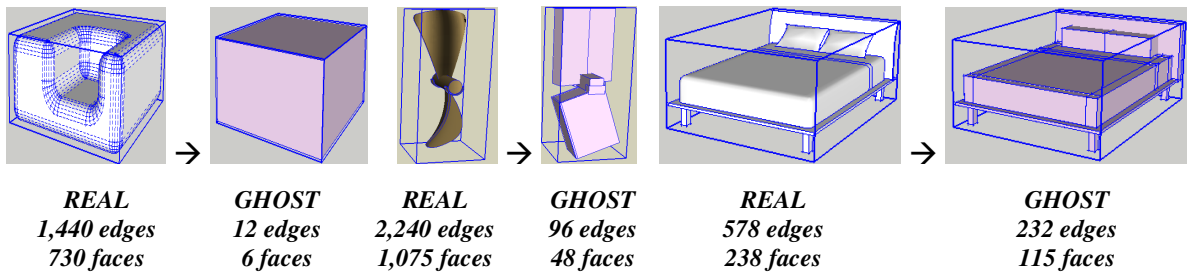
# GHOST COMP

## *Creating and Switching Ghosts for Components*

### QUICKCARD – v1.0

#### 1. Overview

The principle of **GhostComp** is to let the user generate a **simplified** copy of a component (called “*Ghost*”) and then easily switch back and forth between the Ghost version and the **Real version in the model**. The ‘simplified’ ghost version can for instance be just the box shape made out of the bounding limits of the components, so only 6 faces<sup>1</sup>. The Ghost generation is made hierarchically, so that the top level ghost of a component would be made of the ghosts of its sub-components.



The original purpose of the script is actually to **help manipulated complex models**, with lot of components and geometry, where Sketchup may show performance problems.

- The usual workaround to the high-poly syndrome is to organize the model in layers, turning visibility on and off. In practice however, the layer organization may be driven by other considerations than component complexity and mixing both objectives may be cumbersome.
- A complementary workaround is to edit component with the option “*hide rest of model*” turned on. But you may need to use the rest of the model for inference while editing the component.

In contrast, **GhostComp** substitutes less poly-greedy versions of selected components so that the model manipulated on the screen is lighter in number of polygons. When you have complex components, especially if duplicated in many instances, but you are not working currently on them, it may be an acceptable compromise to show them in a simplified version, if it improves the overall performance.

**IMPORTANT:** GhostComp only applies to SU Components, **not Group**, because only Components can be switched quickly and safely in the model. Dynamic components are supported, with some restrictions (see last section).

At this stage, I have no idea whether GhostComp would fit in any practical workflow for designing in SU. So, I just describe here what it does and users will tell.

**NOTE:** GhostComp **NEVER** modifies the original components, even to add an attribute. Likewise, the script will never purge their definition (see section “*Under the Hood*” for more details).

<sup>1</sup> The script offers other options for Ghost generation as the bounding box. See later in the document.

## 2. Installation

**NOTE:** Although the script technically works in Sketchup 6, it is advised to use it with **Sketchup 7** preferably: Only SU7 refreshes the Component browser window and allows trapping Component Edition actions.

- The plugin requires **LibFredo6 v3.3 or higher**, if you do not have it installed already.
- **Unzip 'GhostComp 1.0.zip'**, by doing a *Select All* and then *Extract in the Plugins directory of Sketchup*, answering YES to all prompts for file overwriting, in order to make sure you reconstruct the folder hierarchy. NEVER change the name of files or move them from their folder. The footprint should be (Windows environment):

Name	Type
c:\program files\google\google sketchup 7\plugins\	Folder
GHOSTCOMP_Dir_10	Folder
LIBFREDO6_Dir_33	Folder
LibFredo6.rb	RB File
ZLoader_GhostComp.rb	RB File

**On Mac,** You may have the choice for the directory for unzipping, but **it must be the same as where you installed LibFredo6:**

*Macintosh HD/Library/Application Support/Google SketchUp 7/SketchUp/plugins*

or *Macintosh HD/<user>/Application Support/Google SketchUp 7/SketchUp/plugins*

- **Sketchup versions:** SU6 and SU7. It is advised however to use SU7, as it is faster and also has a few functional benefits.
- **Language:** Initially *English* and *French*<sup>2</sup>
- **Menus:** by default, installed in **Tools > Fredo6 Collection > GhostComp**
- **Icon toolbar:** by Default **'Ghost Comp'**. 10 icons are available. You can however configure the plugin so that fewer icons are shown, while keeping all commands still available in the menu.



- **Default Parameters:** Plugin configuration and specific parameters, via dialog box (see below).

<sup>2</sup> Kind contributors can generate the translation in other languages via the menu 'Translation' and publish the text file (extension .lang) on Sketchucation forum.

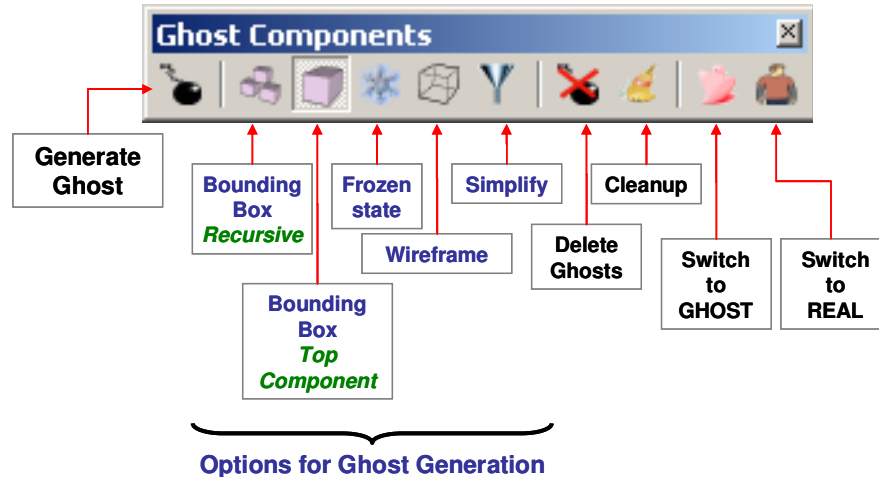
### 3. Using GhostComp



Typically, you select one or several components and then **Generate a Ghost version**, using one of the **5 modes** available. Note that if the component already has a Ghost, it will be replaced.



Then you will **freely Switch to Ghost version or to Real version** selected or all components in the current model view. You can of course continue working on the Real version as usual.



#### 1) Modes for Generating Ghosts

In version 1.0, there are 5 modes for generating Ghosts:

- Simple Bounding Box (non recursive)
- Cascading Bounding boxes
- Wireframe
- Simplify (keep only few faces)
- Freeze current state

Each mode is tailored to a specific situation. Different components in the model can have Ghosts generated in different modes.

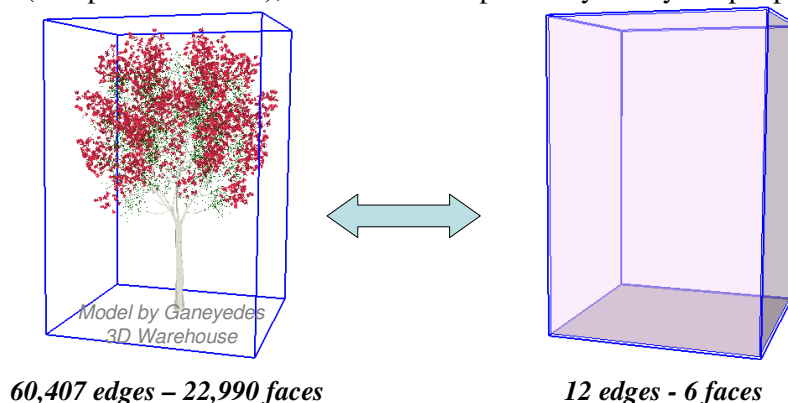


##### Simple Bounding Box (non recursive)

The Ghost is generated as the Bounding Box of the **top-level** component. There is no Ghost generated for the sub-components if any. This mode is very inexpensive in terms of faces and edges since the generated shape will be made of 6 faces and 12 edges.



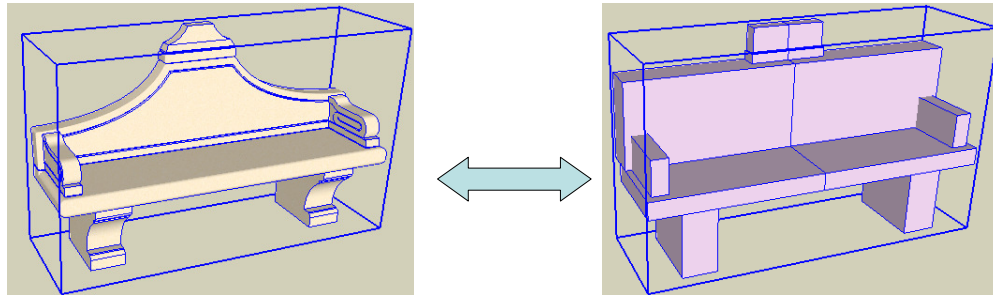
This mode is well targeted for very complex components, such as 3D trees or full sub-model (like part of a house), which can be replaced by a very simple placeholder.





### Cascading Bounding Box (recursive)

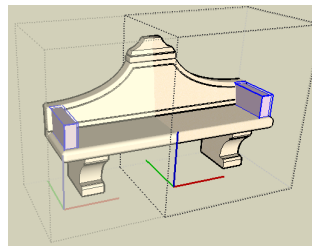
The Ghost is generated as the Bounding Boxes of the **top-level component(s) and its sub-components**, organized with the same hierarchy. This mode is still inexpensive in terms of faces and edges, but usually renders better the overall shape of the component.



7,218 edges – 3,222 faces

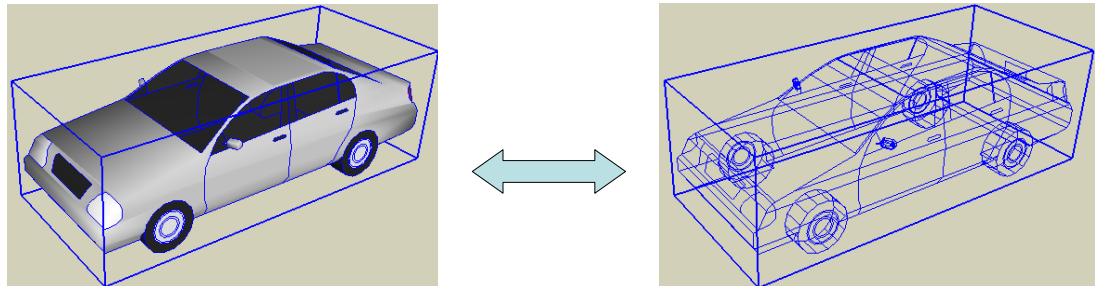
144 edges - 72 faces

Also, Ghosts are generated for all sub-components, so that you can selectively switch between Ghost and Real versions *within* the component itself.



### Wireframe

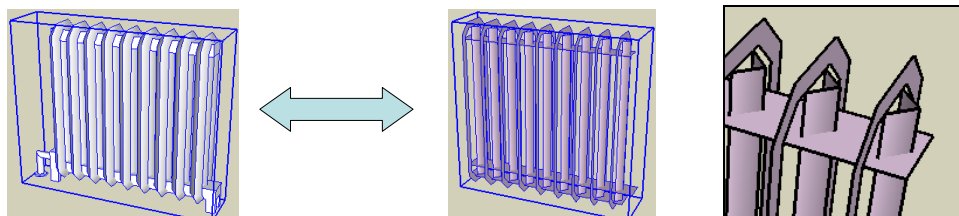
The Ghost is generated as the wireframe of the component and its sub-components, organized with the same hierarchy. The main benefit of the Wireframe mode is to keep in the Ghost all the remarkable points for inferencing with the component.



### Simplify

The Ghost is generated from the original component and subcomponents by keeping only a defined number of faces with the largest area. All other faces and edges are removed.

This mode can be useful to keep the ‘meaningful’ shape of the component, while limiting the number of drawing elements (faces and edges).

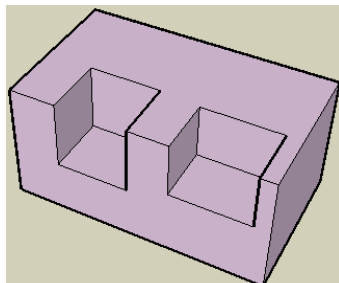




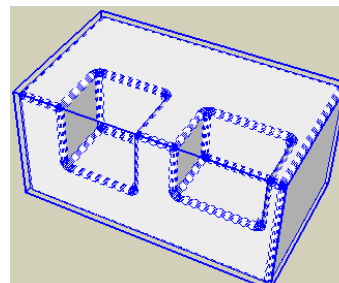
### Freeze Current State

The Ghost is generated as a copy of the original component and sub-components, with just colorizing.

The usual field of application is to take a snapshot of the component before making it more complex, for instance by using Round Corner<sup>3</sup> or Subdivide & Smooth, and more generally before creating detailed geometry.



*36 edges – 14 faces*



*2,172 edges – 1,094 faces*

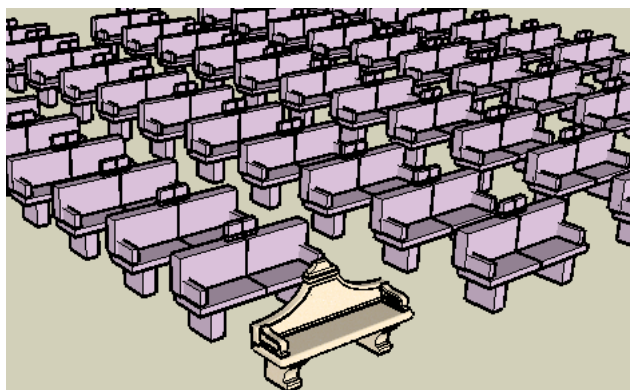


### **2) Switching between Real and Ghost version**

When Ghosts have been generated, you can easily switch between the Real and Ghost version.

This will apply to the selection only if any, or to the whole model, so that you have a better control.

For instance, if you 100 copies of the same component, you may want to turn all of them to their Ghost version, and then switch back only one to Real so that you can work on it without the payload of showing all 100 in the model view.



**Note:** Switching between Ghost and Real version is contextual to the Active model. If you are within a component, then the operation applies to the components present within this component. And if you are now outside the component, the switching won't apply. To switch to Real version recursively, you need to **double-click** on the icon.



### **3) Deleting Ghosts**

You can delete the Ghosts of components by selecting them and clicking on the icon. If there is no selection, then you will remove all Ghosts from the model. In any case, all components in Ghost version will be switched to their Real version before deletion of the Ghost.

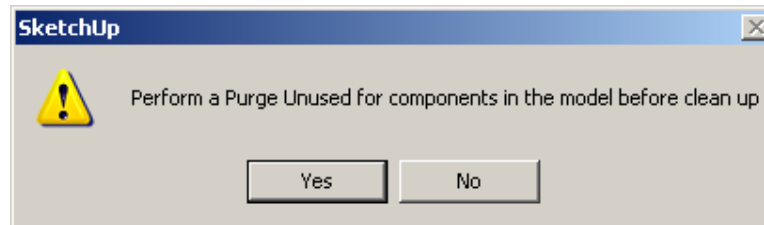
<sup>3</sup> The whole idea of GhostComp actually came from a remark by Pixero about the wish to delay as much as possible the Round Corner operation until rendering.



#### 4) Clean up

When you delete components in the model, their Ghost version is not immediately deleted (in particular because you may perform an undo). An automatic clean up is performed when you first use a Ghost operation (Switch for instance) after you first opened the model.

You may want however to do it manually at any time by clicking on the Cleanup icon. It is strongly advised to do first a save of the model. The operation ask whether you wish to proceed with a 'Purge unused' components (to remove actually the definition of deleted components from the model) and then perform the clean up.



#### 4. Under the Hood

When you generate a Ghost for a Component, the following happens in the back:

- 1) GhostComp creates a **special Ghost layer** named `__GHOSTCOMP_LAYER__`. It is hidden by default as well as all component instances that will be put in this layer.
- 2) A hidden copy of the original component is put on the Ghost layer. This is to prevent that a Purge Unused erase the original component when in Ghost version in the model.
- 3) The Ghost component is created from a copy of the original component, followed by a 'Make unique', and then modified according to the selected generation mode. Colorization of the Ghost is also performed with a special material called `__GHOSTCOMP_MATERIAL__`. A hidden instance of the Ghost component definition is put on the Ghost layer. The definition name of the Ghost is based on the prefix "`zzGHC_`" followed by the name of the original component definition.
- 4) When the original component has sub-components, the operation is done recursively (except for the Simple Bounding Box mode).
- 5) GhostComp attaches special attributes to the component *instances* created on the Ghost layer (dictionary "`__GhostComp__`") so that it can maintain a cross reference between Real and Ghost component definition.
- 6) You can observe the Ghost component definition in the Component browser (unfortunately, it is not possible to hide them with the Sketchup API).






A few things to keep in mind:

- **GhostComp does NOT touch or delete the original component definitions.**
- **The cross-referencing does not depend on the external names of the components,** which you can thus freely change.
- **When you regenerate a Ghost on a component that already has one, the old Ghost is replaced.** There is only one Ghost version per component definition.
- **You should try to avoid making the Ghost layer visible.** Actually, it is not very nice, because all component instances there are piled up.
- **You should avoid editing a Ghost.** A warning message is displayed if you do. This is mainly to prevent that you do real design work on the Ghost by accident, instead of doing it on the Real version (note however that there is actually nothing wrong in the algorithm if you modify manually a Ghost, it will simply be used).

## 5. Default Parameters

You can set persistently some parameters across Sketchup sessions via the Default Parameters dialog box.










### 1) Parameters and Options of GhostComp

<i>Parameters at startup of Sketchup</i>	
Default option for creating Ghosts	Option: SIMPLE BOUNDING BOX 
Max number of faces to keep with option SIMPLIFY	10 
Color for rendering ghosts	Thistle 
Use observer (recommended for warning edition of ghost components)	<input checked="" type="checkbox"/> 
Switch automatically to Ghost version after generation of Ghost	<input type="checkbox"/> 

Note that the Ghost color can be changed live in the model, but there is only one for all Ghosts, as it is actually a material in the model.

### 2) Configuration of the Plugin

You can also configure the integration of the plugin in Sketchup (toolbar in particular):

<i>Plugin Configuration</i>	
Folder for Alternate Icons, cursors and other images	<input type="text"/>   
Toolbar name	Ghost Components 
Icons visible in the toolbar	<input checked="" type="checkbox"/> Generate Ghost for the selection  <input checked="" type="checkbox"/> Option: CASCADING BOUNDING BOX   <input checked="" type="checkbox"/> Option: SIMPLE BOUNDING BOX 
Top Menu	Fredo6 Collection (Sketchup - Tools) 

At installation, all icons are shown. In practice, you may want to keep only 3 icons: one for Generating Ghosts, and the two others for Switching to Ghost and Real versions. All commands remain anyway available via the Menu.



## 6. Limitations – Known Issues

The concept implemented by GhostComp has several side effects due to the limitation of the Sketchup Ruby API. This is why I think it should rather be developed in the native Sketchup application by the Google SU team.

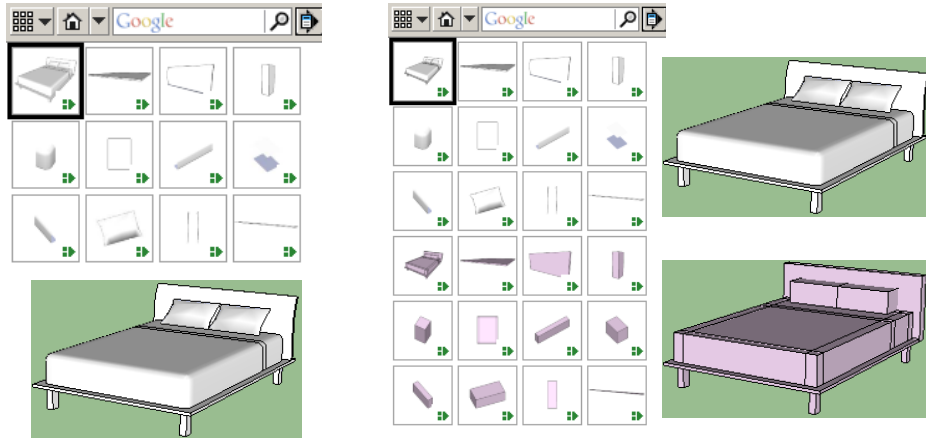
### 1) Performance

The stress has been put on the speed to Switch between Real and Ghost versions, as this is an operation that can be performed frequently, even on large models.

The Ghost generation may be much slower, as it deals with copying components, and modifying them, including colorization. This is normally done less frequently. In any case, there is a progress bar, but the operation cannot be interrupted in the current GhostComp version.

### 2) Component Browser

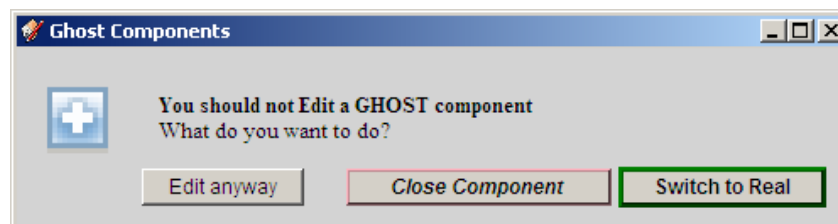
As seen, GhostComp creates Ghost versions for each component and sub-components. They are all visible in the Component browser window for the model. This may indeed make your component environment more crowded than what is really meaningful for the design. In SU7, the thumbnails are normally updated. In SU6, there are not (this is a limitation of SU6).



Another issue is related to the reduced performance when the component browser is opened on the current model (this is true for most plugins). In SU6, this may generate abnormal delay when you generate a Ghost. In SU7, this is better managed however.

### 3) Editing a Ghost

The script displays a warning message whenever you try to edit (i.e. open) a Ghost component. In SU7, this action is intercepted whether you double click or use the contextual menu. In SU6, only the double click is intercepted.



Also, although the message box proposes to switch to the Real version, you will have to double-click thereafter (there is no programmatic way to edit a component from Ruby).



## 7. Programming API (for Script developers only)

The script offers a programming interface for Switching between Real and Ghost versions.

IMPORTANT: there is no `'require'` statement for GhostComp. You need to check dynamically if GhostComp is installed, with a test on `defined? (GhostComp)`.

- Switching to Ghost: `GhostComp.switch_to_ghost(lst_comp=nil)`
- Switching to Real: `GhostComp.switch_to_real(lst_comp=nil)`

...where `lst_comp` is an Array of component instances (actually a list of any entity). The methods will ignore entities which are not component instances or do not have a Ghost attached. Furthermore, if for instance you call `switch_to_ghost` on a component which is already in Ghost version, nothing will happen.

If you omit the argument `lst_comp` or pass it as `nil`, then the operation will apply to the whole active model.

The Return value is `true` if the selection contained valid component instances, and `false` otherwise.