Working with Virtual Components

Tips for ensuring changes are saved

September 2012

Revision 1.0





|  |
| --- |
|  |

**Table of Contents**

[1. Introduction 4](#_Toc335320319)

[2. Understanding Virtual Components 4](#_Toc335320320)

[3. Working with Lightweight Virtual Components 4](#_Toc335320321)

[4. Working with Read-Only Virtual Components 6](#_Toc335320322)

[5. Visual Symptoms 8](#_Toc335320323)

**Revision History**

|  |  |  |
| --- | --- | --- |
| **Rev #** | **Date** | **Description** |
| *1.0* | *Sept. 13, 2012* | *Document created.* |

1. Introduction

The objective of this document is to share information related to how virtual components work and the various workflows that could lead to virtual components becoming externalized unintentionally or making changes to virtual components that are not retained when closing and reopening the parent assemblies.

1. Understanding Virtual Components

When a SolidWorks assembly contains a virtual component, a temporary physical file is created in the user’s temp directory, for example **<...\AppData\Local\Temp\*1\swx7640*\VC~~\*Assem1*\...>.** When the parent assembly is closed, the temporary virtual component files are removed from the temporary directory. Within the same SolidWorks session, the same ***\swx7640\*** is used; if a new session of SolidWorks is started that session will use a new path such as ***\swx3896\***. When first inserted, a virtual component is displayed in the feature tree with brackets **[ ]** around the name. However, the name can be changed to remove the brackets so this is not a reliable means of identifying virtual components. It is important to understand the changes to a virtual component are saved in the parent file, not in the virtual component itself as the virtual component is a temporary file created from the stored information in the parent file. Listing the file references for the parent assembly will show the reference path of virtual components as **<save internal to assembly>.** A **RMB** (right mouse button) on a virtual component gives the option to **Save Part (in external file)**.

1. Working with Lightweight Virtual Components

When a sub-assembly is opened lightweight, SolidWorks does not look at the status of its sub-components during the save operation. Therefore, if a sub-component has been modified, it is possible to save the main assembly and not save the changes to the lightweight sub-assembly which will result in the changes to the sub-component being lost.

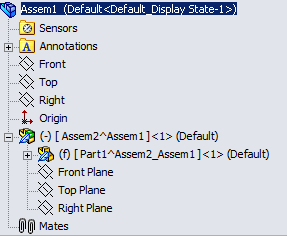


Figure 1: In this example "Assem1" is open for write access. "Assem1" is a physical file and "Assem2" and "Part1" are virtual components.

After completing the changes to a virtual component, when the virtual component is closed and SolidWorks returns to the parent assembly, the first dialog that will be displayed, unless the user has chosen to permanently dismiss the message, is one that indicates that changes have been made and asks the user if they would like to rebuild the assembly.

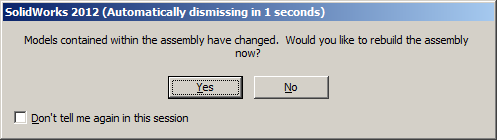


Figure 2: Rebuilding the parent assembly is suggested to help ensure that SolidWorks is aware of all pending changes and can properly warn the user of files that require a save.

If the user selects **No** then the sub-assembly will remain lightweight and will not indicate that the main assembly is dirty. If the user also checks the **“Don’t tell me again in this session”**, the user will no longer be warned of the need to rebuild putting that user at risk of unknowingly losing changes to virtual components.In this case there will be no indication that a save is needed.

Upon selecting either Save or attempting to close the file SolidWorks will present a dialog asking to if the user wants the document to be rebuilt before saving. Again this dialog can be dismissed by checking “Don’t tell me again in this session”, resulting in the user not being aware of the need to rebuild before saving. If the user does not select to rebuild, SolidWorks will not be made aware that the lightweight sub-assembly had a modified component and those changes will be lost.

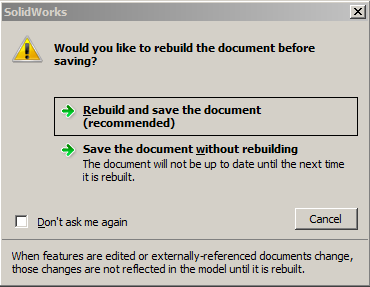


Figure 3: This dialog is displayed during save or close of a document containing changes to references that have not yet been rebuilt in the main assembly. Selecting the save without rebuild with the “Don’t ask me again” option checked will result in assemblies be saved out of date without the user knowing.

1. Working with Read-Only Virtual Components

The write access of a virtual component is based on the write access of the parent file which is where the virtual components information is stored. When opening a virtual component in its own window to make changes, the temporary physical file that is opened is not read-only, therefore the virtual component will appear to have write access even though the parent file is still read-only. Changes saved to this temporary virtual file will not be saved in the read-only parent file although the changes will still appear when returning to the parent assembly.

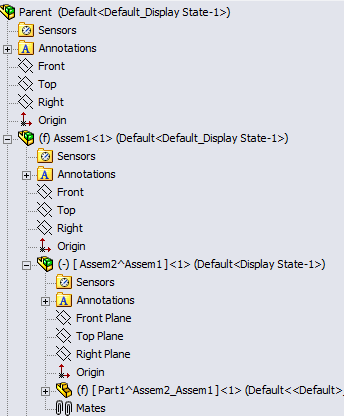


Figure 4: In this example the "Parent" file has write access and the sub-assembly "Assem1" is read-only. "Parent" and "Assem1" are physical files and "Assem2" and "Part1" are virtual components.

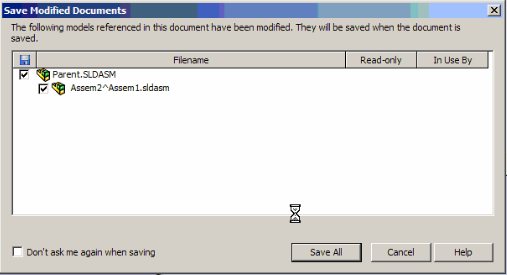


Figure 5: This dialog will not indicate that the virtual sub-component is read-only which can lead to the false impression that the component is being saved and changes can be lost. Dismissing this message by checking “Don’t ask me again when saving” can result in the user not being unaware that a virtual sub-component requires a save and changes can be lost.

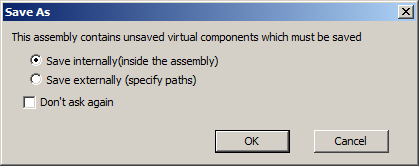


Figure 6: This dialog is shown when saving an assembly that has unsaved virtual components. Choosing to save externally and dismissing the message by checking “Don’t ask again” can result in virtual components becoming externalized unintentionally.

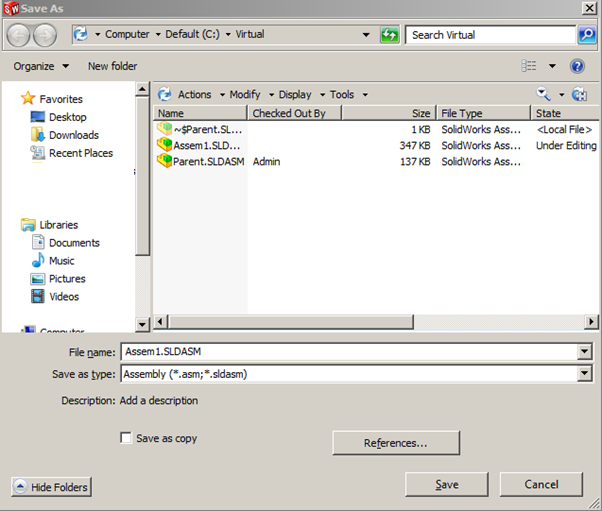


Figure 7: The SolidWorks option “Don’t prompt to save read-only referenced documents (discard changes)” shows this dialog when disabled and saving at the Parent assembly. The prompt is to save the sub-assembly Parent which is the physical parent file of the dirty virtual component.

1. Visual Symptoms

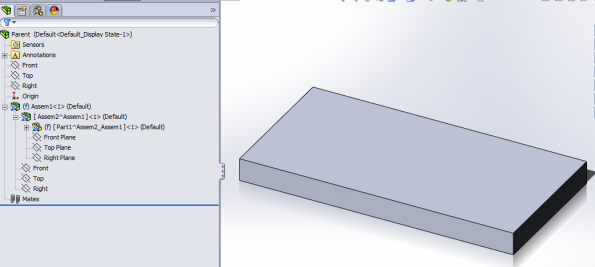
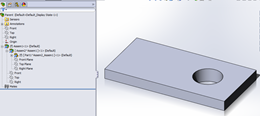


Figure 8: The changes will appear in the preview image displayed during open as this reflects the tessellation data stored in the parent assembly. After the open is finished the preview is replaced with the actual data image.