**SolidWorks to URDF Exporter**

The SolidWorks to URDF exporter is a SolidWorks add-in that allows for the convenient export of SW Parts and Assemblies into a URDF file. The exporter will create a ROS-like package that contains a directory for meshes, textures and robots (urdf files). For single SolidWorks parts, the part exporter will pull the material properties and create a single link in the URDF. For assemblies, the exporter will build the links and create a tree based on the SW assembly hierarchy. The exporter can automatically determine the proper joint type, joint transforms, and axes.

|  |  |  |
| --- | --- | --- |
| [Download Installer](https://github.com/ros/solidworks_urdf_exporter/releases) | | |
|  |  |  |

**SolidWorks Version Compatibility**

There is a known STL export bug with SolidWorks 2018 that exists up to Service Pack 4 that renders this add-in unusable. If you are using 2018, please update to service pack 5 or use SolidWorks 2019 or later. 2017 and below may also work

**Some Important Items**

Development on this plugin as of recently has come from the generous donations of several ROS community members. Without this support or from pull requests from communities members, development would not be able to continue. We also appreciate any bugs or feature requests so that we can work on them when the resources become available.

This project is not dependent on ROS and can be used for exporting SolidWorks files for any URDF needs. The URDF will include rospack URI file locations ('package://'), so you will need to change those for non-ROS systems (some Gazebo systems can handle 'package' URIs).

This add-in has been tested on Windows 10 64bit with SolidWorks 2018 SP 5 64bit. It currently does not install on 32bit machines. The add-in makes use of the most up-to-date SW API so it may not work with versions earlier than SolidWorks 2018 (but this hasn't been confirmed).

To learn more about URDF, please review its [documentation](http://wiki.ros.org/urdf).

If you have questions about the SolidWorks to URDF Exporter after reading this documentation please check [ROS Answers](http://answers.ros.org/) for possible answers. If you don't find your answer, please submit a question with the tag 'sw\_urdf\_exporter'.

The code has been migrated to Github from its original BitBucket page. Use Github's

* [issue tracker](https://github.com/ros/solidworks_urdf_exporter/issues).

Details of bug fixes and new features are put in the [Change Log](http://wiki.ros.org/sw_urdf_exporter/change_log).

**Installing**

The download is available as a pre-compiled installer or as source as a Visual Studio C# project.

This sounds made up, but you cannot install the files into a directory labeled "SW2URDF". For some reason, Windows struggles to find the supporting plug-ins, and will throw an unhandled exception error. If anyone has some suggestion as to why, I'd love to know.

**Pre-compiled Installer**

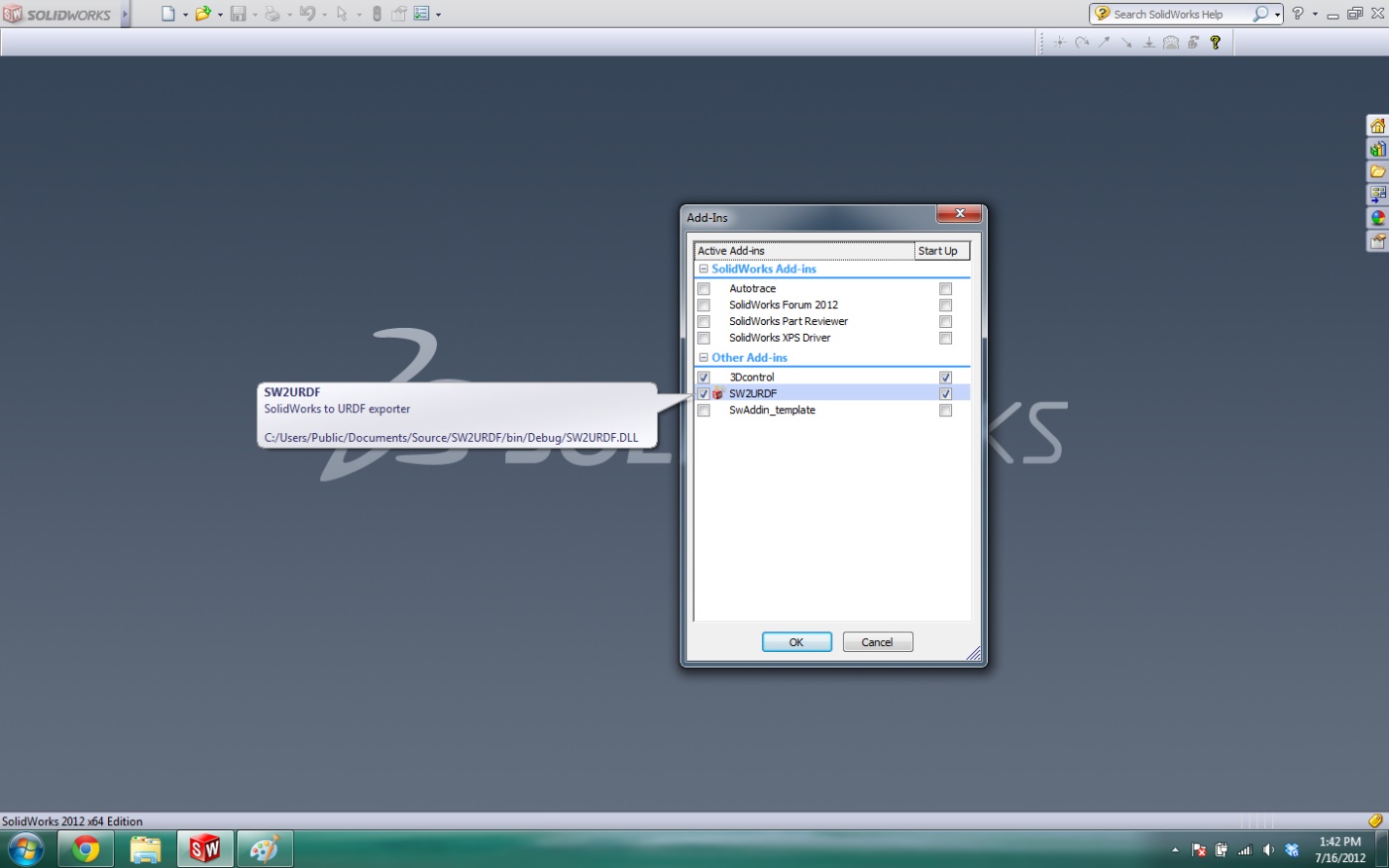
This installs the most current stable release.

1. Upgrade to at least [.NET Framework V4.7.2](https://dotnet.microsoft.com/download/dotnet-framework) if you don't have it already.
2. Download the installer linked above.
3. Run the installer. It will ask for permission to make changes to your computer. Your Windows account will need to have sufficient privileges to make these changes. It should install into the SolidWorks directory in program files.
4. Open SolidWorks, under "Tools>Add-ins" you will see a SW2URDF item at the very bottom. If this setup file doesn't work, try installing manually as described below.

**Compile source**

These are the most up-to-date source files. Because this is checked in regularly, there may be bugs or un-finished code in the source files. If you need a stable version of the source, check-out the revision tied to the installer.

1. To build the source, you'll need to download [Visual Studio Community](https://visualstudio.microsoft.com/vs/community/).
2. Checkout the [source](https://github.com/ros/solidworks_urdf_exporter) with git.
3. Run Visual Studio as administrator. Your Windows account needs to have these privileges to register the .dll. To do this, right click on Visual C# in your start menu and click, 'run as administrator'.
4. Click on "Debug>Start Debugging" or press F5. It will compile and run SolidWorks together. After doing this once, you only need to run from Visual Studio to re-compile. Otherwise, to just use the tool, you can start SW normally.
5. When SolidWorks is open, under "Tools>Add-ins" you will see a SW2URDF item at the very bottom. If this setup file doesn't work try installing manually as described below.



The exporter has two separate forms, one for exporting single SW parts as a single link and the second for exporting assemblies as a set of links and joints.

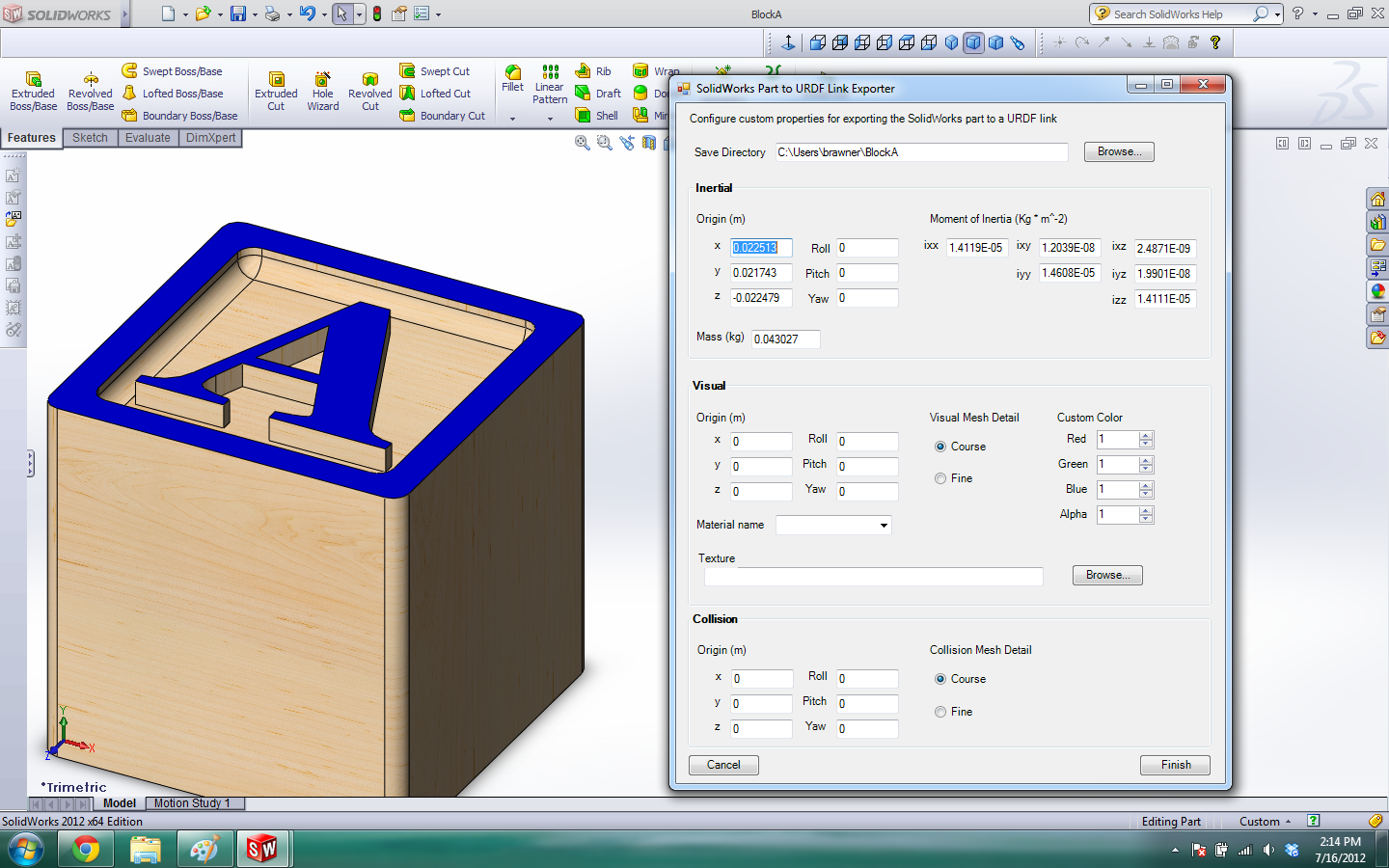
# Export a SolidWorks part to URDF

**Description:** This tutorial describes the very simple process for exporting a SolidWorks part to a URDF.

### Exporting a part

1. Open your part in SolidWorks.
2. Click "File>Export to URDF" (sometimes its "Extras>File>Export to URDF")

This will bring up a single page form that loads the material property, color property and mass property information into the form. If you desire, you can change any items you like, but the part will be loadable into Gazebo or RVIZ without any modifications.



The default save path is a package named after the part in your home folder, click browse to change this.

You can add a texture to the link by selecting the texture from your computer. This file will be copied into your URDF package.

Click finish to create the package. It will export the meshes, copy the textures and create a URDF that specifies the locations of these files with a 'package://PackageName' relative location. If you do not have rospack installed on your target computer, you will have to edit these locations manually.

### The built package

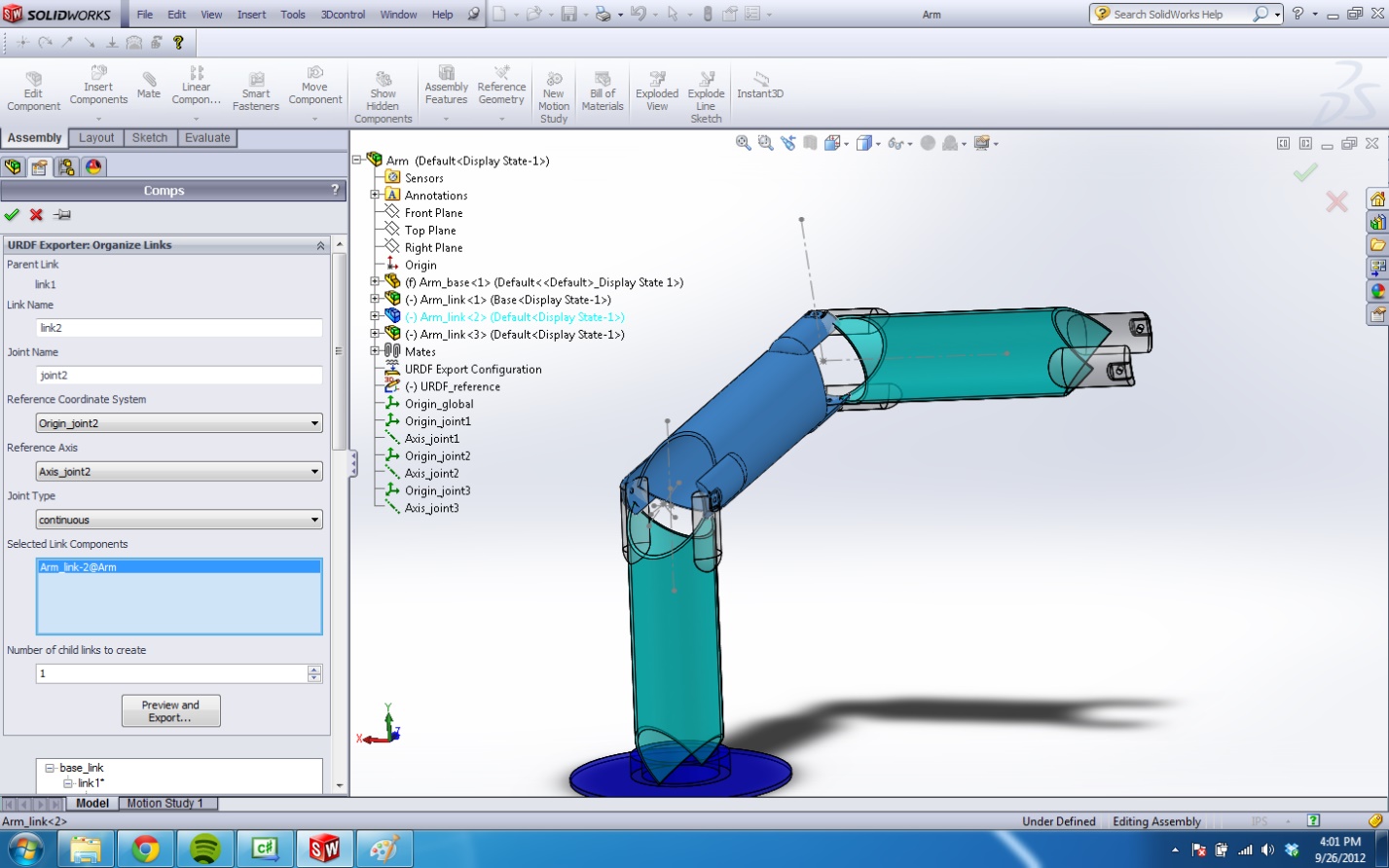
The package will contain directories for meshes, textures and robots. It will also contain a ROS package.xml (manifest) file so you can use this as a ROS package by just copying it to your ROS system. The path locations in the URDF are relative to the package itself. If you do not have [rospack](http://www.ros.org/wiki/rospack), you will need to change the paths manually so that they refer to locations on the machine using the URDF files. If you do use rospack, you will need to add the package to a location in your $ROS\_PACKAGE\_PATH. To simplify complicated meshes, you may like to use a program like [MeshLab](http://meshlab.sourceforge.net/) or [Blender](http://www.blender.org/).

# Export a SolidWorks Assembly to URDF

**Description:** This tutorial covers the process of exporting a SolidWorks Assembly to URDF using the SolidWorks to URDF Exporter

## Property Manager

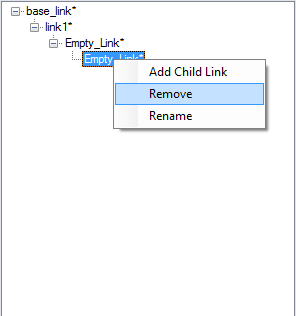
The exporter first brings up a property manager page for you to configure your URDF Export. You will need to configure each link and build the tree manually. After configuring this for the first time, the tool will save your configuration with the assembly itself. You should then be able to skip this step afterwards.



For each link, you need to give it a unique name, give it a unique joint name, select the components (assemblies or parts) in SolidWorks that are to be associated with that link and add the necessary number of children. If you have reference coordinate systems or axes you would rather the tool to use, you may select those from the list. You can also manually specify what each joint type is. Each URDF has only one base link. The joint configurations are disabled for this one link because there doesn't exist a joint to a higher level link. You only need to name the link (if you don't want to call it base\_link), select its components and create its child links.

In SolidWorks, if you select components on the actual assembly display instead of the FeatureManager Design Tree, you will only be able to select parts. More likely you would rather select assemblies that represent the entire link, in which case you will have select them from the Design Tree as pictured above.

Note that for performance reasons, you can only select parts or subassemblies two levels deep into the [SolidWorks](http://wiki.ros.org/SolidWorks) model.



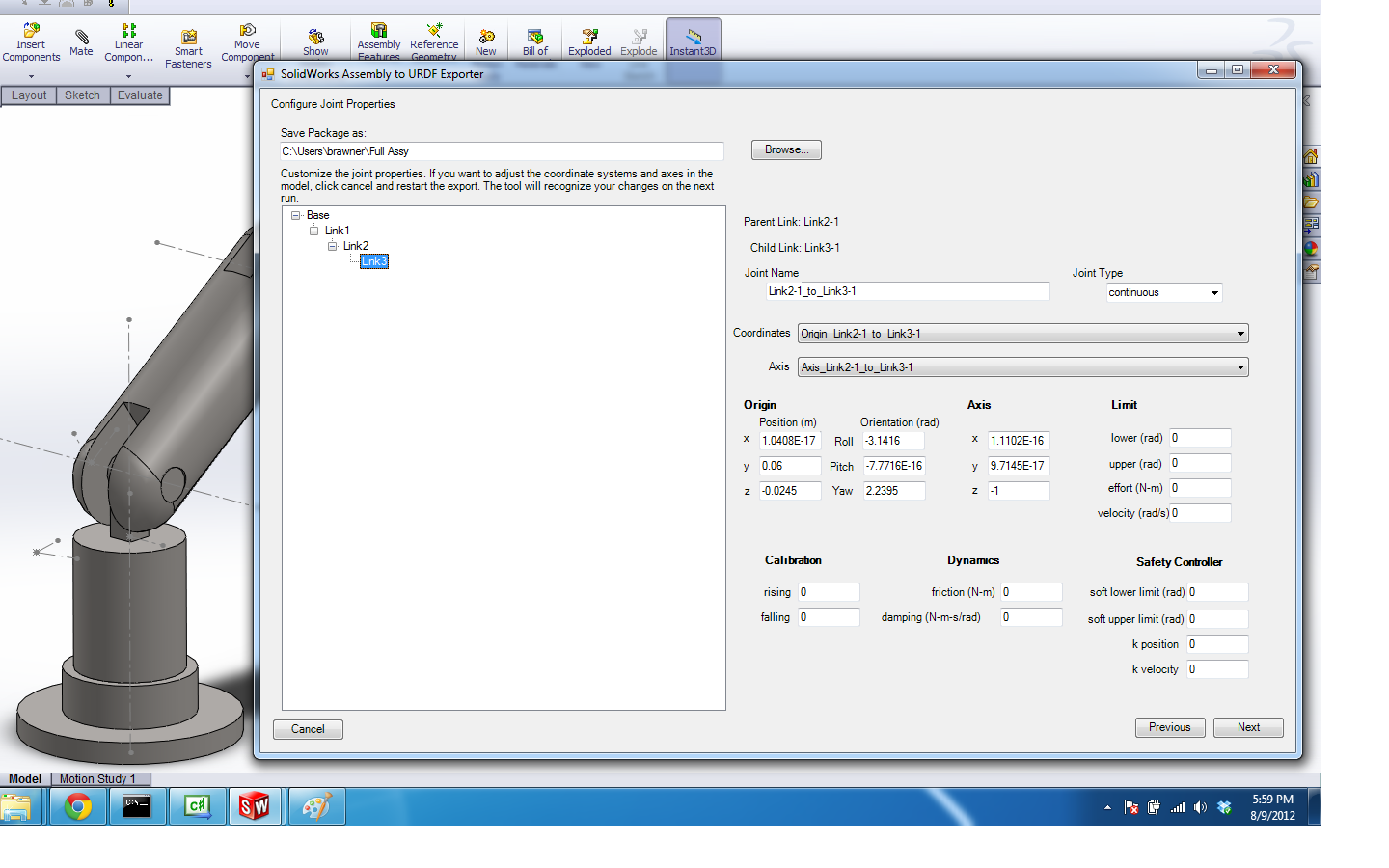
The configuration tree shows you all the links you've added. For each child link on the tree, a joint will be created to its parent link. You can select any link you've already added to change its properties. Right clicking a link will allow you to add children to or remove the link. You can also drag and drop links to re-order them. Dragging a parent link onto a child will cause the child link to switch positions with the parent link. The parent link's other children still stay with the original parent link.

Clicking the green check mark will just save your configuration and exit. Clicking the red x will allow you to exit without saving changes made to your configuration.

When you are ready to build your URDF, click "Preview and Export...". If you have specified the tool to automatically generate coordinate systems or axes, it will build them at this stage.

## Joint Properties

Should you find that these joint origins are often not created in the most desirable location, you can change the reference coordinate systems and axes to suit your needs outside of the exporter. However, the model will work fine built as-is. The 3D Sketch is just used for temporary construction, but you may find it useful for adjusting the locations of the reference geometry. Pressing cancel on the Joint Properties page will allow you to save the names of the coordinate systems and axes to your configuration. You may then proceed to adjust the coordinate systems and axes. Restart the export process by clicking "File>Export to URDF". Ensure that the right names of coordinate systems and axes are saved for each link. The tool will no longer build them and instead refer to the reference geometry already in place.



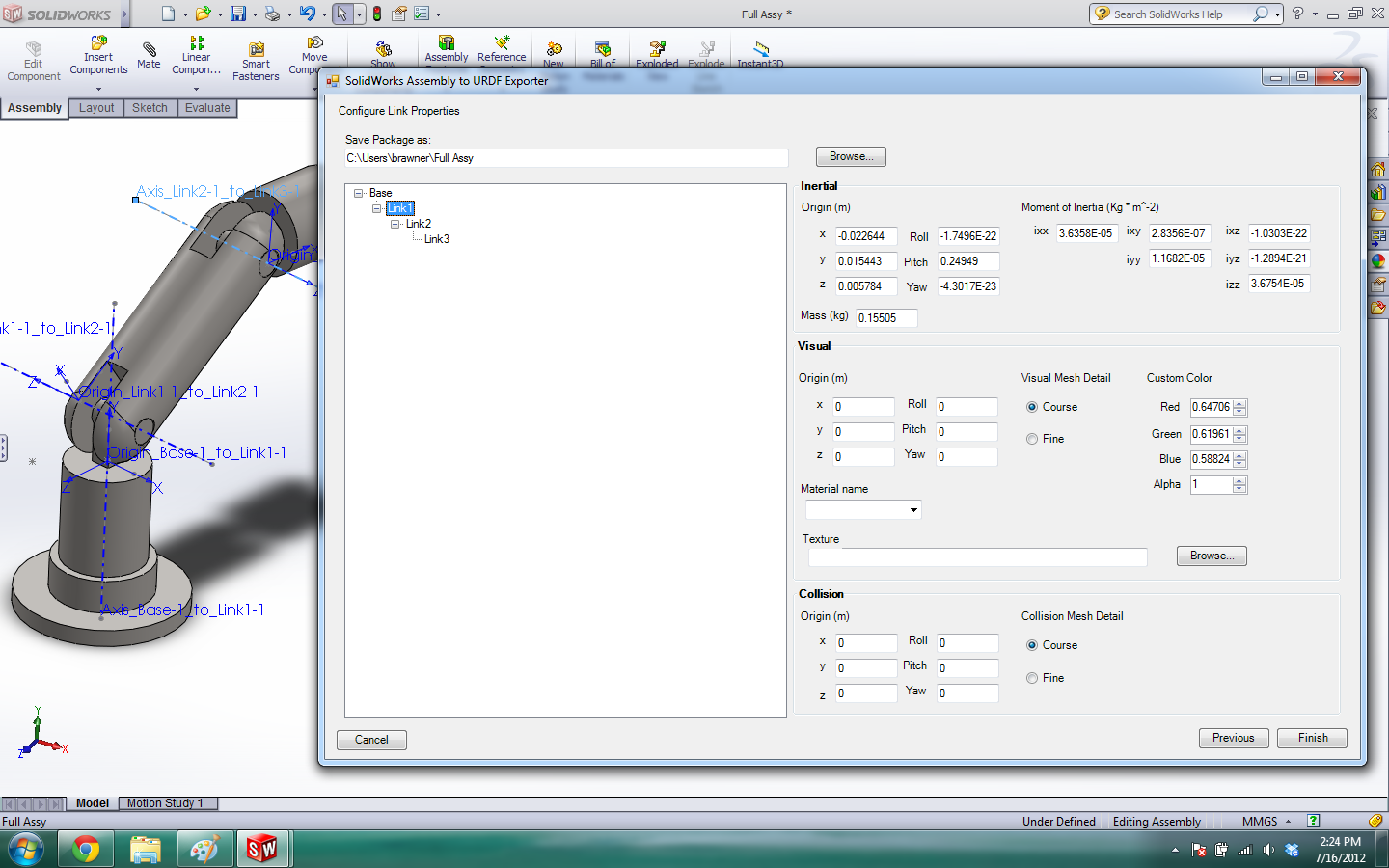
The joint page includes a non-configurable joint tree, where clicking each joint will bring up its properties on the right-hand side. You can customize the properties of any joint before exporting. These properties will not be saved however with the configuration. They will need to be re-entered each time.

Fields that are initially blank are not required by the [urdf](http://wiki.ros.org/urdf) specifications. If they are left blank they will not be written to the URDF file. If you change a property of a section that includes other required fields, but neglect to specify them, they will be filled in with 0.

Click next to go to the Link Properties page

## Link Properties

This page presents a similar view to the Part Exporter discussed earlier. You can change the link properties of any link in your tree. You can add a texture, change the color, change the origins of different sections, change the moment of inertia tensor, mass, etc. These changes will also not be saved with your configuration.



Click Finish to create your URDF package

## The built package

The package will contain directories for meshes, textures and robots. It will also contain a ROS package.xml (manifest) file so you can use this as a ROS package by just copying it to your ROS system. The path locations in the URDF are relative to the package itself. If you do not have [rospack](http://www.ros.org/wiki/rospack), you will need to change the paths manually so that they refer to locations on the machine using the URDF files. If you do use rospack, you will need to add the package to a location in your $ROS\_PACKAGE\_PATH. If you find that your meshes are too complicated you can refer to the [tutorial for improving complicated assembly exports](http://wiki.ros.org/sw_urdf_exporter/Tutorials/Organizing%20SolidWorks%20Assembly%20for%20URDF%20Exporter) or you can simplify complicated meshes by using a program like [MeshLab](http://meshlab.sourceforge.net/) or [Blender](http://www.blender.org/).

# Simplify the SolidWorks and URDF Exporter Workflow for Large Assemblies

**Description:** This tutorial covers how to organize your SolidWorks assembly to create a better workflow and also how to reduce the exported mesh complexity when dealing with complicated assemblies.

# How to organize a complicated SolidWorks assembly for export

Introduction

This tutorial is a collection of all techniques to help make the exporting process simpler.

Goals in simplification:

1. Keep work flow as natural as possible and integrate easily with project
2. Reduce mesh complexity without sacrificing goal #1

Assemble all link components into one assembly

The easiest way to organize your assembly is to have as few components as possible associated with each link. Instead, group as much as possible into a single assembly for every independent link. This suggestion is not due to limitations in the software, but instead it's more likely with a longer list that the associated components will change throughout the development process. So this practice will reduce the amount of configuration necessary before each export. It will also help ensure that the saved export configuration will remain up-to-date for everyone who opens the assembly.

Obviously this won't help if you suddenly decide your robot arm requires 6 degrees of freedom instead of 5, but as you add more components to it it should minimize tinkering around with the configuration.

Setup Configurations and Display States

To avoid exporting the meshes of every single nut, bolt, screw, or flux capacitor in your assembly, you can hide them before exporting. However, solely hiding them doesn't allow you to save this configuration for later exports.

Create a Display State specifically for exporting. Click the configuration tab in the Property Manager and right click the bottom section where Display States are located. Add a new display state and name it something like "URDF Export" so that you and others can return to it later. From the property manager tree, hide all the subassemblies and parts you do not want displayed. Double clicking the original display state, probably 'Display State-1', unhides all the components that are hidden in the URDF Export display state.

Most SolidWorks users are familiar with Configurations, but these only allow you to suppress components or features. This is not useful because the exporter will ignore all those components when calculating mass properties and your mates may break. Display States is the analogous feature but for hiding parts.

SolidWorks allows you to 'Link Display States to Configuration', but this is not recommended. Despite the ability to link configurations in the main assembly with configurations in subassemblies, the Display States aren't inherited for some reason. So you'll have to bite the bullet and work in the top level assembly to hide all the subassembly components. Annoying I know, and it's SW's bug, not mine. I'm open to suggestions for how to deal with this.

Skin Parts

SolidWorks does not have great tools for reducing the triangle count in a mesh below their 'course' export option, which is never course enough for simulation/collision detection. Therefore when not using the Exporter, many are forced to create their own skin meshes and incorporate them into the URDF. Since our goal is to eliminate work outside of the natural work flow specifically for exporting, it's recommended to create your skin part in SolidWorks.

Create a part that envelopes your entire link and that vaguely resembles its shape. Set the material on this skin part to 'Air'. It won't matter, but you might want to change the appearance. Insert it into your main assembly. Place this component over your link in the approximate location and create a 'lock' mate to another component in your assembly. Then activate your 'URDF Export' Display State and hide every component but your skin parts. Then in your default 'Display State-1' or whatever its called, hide the skin parts.

Home State Mates

If you have a preferred home state for the robot, it is recommended to maintain mates for each degree of freedom. Name these mates something recognizable (like 'Shoulder Home State') so that it is a simple manner of suppressing them for the first export process. Once the reference geometries for the joints have been created and the joint type has been saved, then you do not need to suppress the mates each time. The tool will refer to its configuration instead of trying to automatically generate them, which would be inhibited anyway by the unsuppressed home state mates. However, you should ensure the coordinate systems and axes are located properly, and their names are saved within the configuration.

Combining Collision and Visual URDF

Note that when creating both visual and collision meshes, the SW plugin must be run twice. Two URDFs will then be generated - one using visual meshes and one using the custom collision meshes you created. Merging these two URDFs is still a manual process (not supported by the SW URDF plugin) but it is not difficult. Simply copy the <collision> elements into the correct location in the master URDF.

Source :

<http://wiki.ros.org/sw_urdf_exporter>

<http://wiki.ros.org/sw_urdf_exporter/Tutorials>