

CP9911

Get your Inventor D(e)river License

Jeff Stueck

IMAGINiT Technologies

Description

What would you do if Autodesk, Inc., removed the *Save As* command from the next version of Inventor software? Once you're familiar with Inventor software's deriving functionality, you probably wouldn't even notice if the *Save As* command went missing. During this class we will explore what the *Derive* command is, why you would want to use it, and how to use it effectively. (Spoiler Alert: it's really pretty simple, once you understand how it works). This class is for you, if you're already comfortable with Inventor software 3D modeling, but want to be more efficient, accurate, and flexible in your workflow. Let's get your Inventor D(e)river License.

Learning Objectives

- Explain the differences (and similarities) between *Derive* and *Save As*
- Describe which objects can be Derived and how they may be linked together (*Pushing* or *Pulling*)
- Identify where *Derive* can be used in your workflow (and where you may already be using it)
- Perform the steps of Deriving using:
 - Derive Command (from Part Mode)
 - Make Components/Make Part (from Part Mode)
 - Shrinkwrap (from Assembly Mode)
 - Copy Part (from Part Mode within the context of an Assembly)

Your AU Expert

Jeff Stueck is an Application Expert at IMAGINiT Technologies where he specializes in software solutions and workflows relating primarily to the needs of manufacturers.

Jeff grew up in a machine shop during a time when the manufacturing industry was just beginning to benefit from computer technology, so his skills and experience evolved naturally along the way.

In college, Jeff studied Accounting, which eventually led him back into manufacturing and technology as he developed and supported several businesses as an industry accountant.

Most notably, Jeff owned and operated a high-end custom furniture design and manufacturing firm in Park City, UT for nearly 10 years, and then played a key role in the development and growth of the cabinetry division at Woodland Furniture, in Idaho Falls.



The differences (and similarities) between *Derive* and *Save As*:

Save As - Creates a copy of the currently active file so you can:

- Make changes to the new file without affecting the original
- Revert back to the previously saved file if needed

The Save As Command was the original way to manage multiple versions of digital files saved at a certain point in time.

Potential problems with Save As: Just ask anyone who has tried (or is currently trying) to manage file versions manually; especially if there are multiple users creating versions and sharing files.

Derive - Similar to Save As because it creates a copy, but using Derive also links the copied (Derived) object back to the original file:

Dictionary Definition of Derive:

- To obtain something from a specific source
- To know the origin of something

Additional Meaning for Inventor Derive:

- Creates a link back to the source
- Changes made to the source (Base Component) affect everything linked back to it

The Derive function is free from the constraints of time. Not only can you trace Derived Stuff back to its origin, but you can make changes in the past which will affect everything from that point forward.

When should you use Derive vs Save-as?

When to Derive:

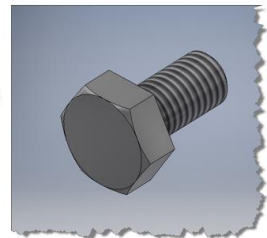
The *shape, size, or configuration* of the model is likely to change



Especially if it will change (or be changed) relative to another file

When to Save As:

The *shape and size* of the model is static with no chance for change



You don't need additional visibility into the origin or history of the model



The Stuff we can Derive:

It's hard to name everything that can be *Derived* because the options are so diverse, so I'll just use the word *Stuff*.

The following image shows a portion of the *Derived Part* window, but there are additional methods (addressed later in this document) for *Deriving Stuff* without passing through this window.

During this session, we will focus on Deriving Solid Bodies and Surface Bodies, but Deriving the other Stuff is very similar as well.

What kind of "Stuff" can be Derived?

Solid Bodies

Surface Bodies

Sketch Blocks

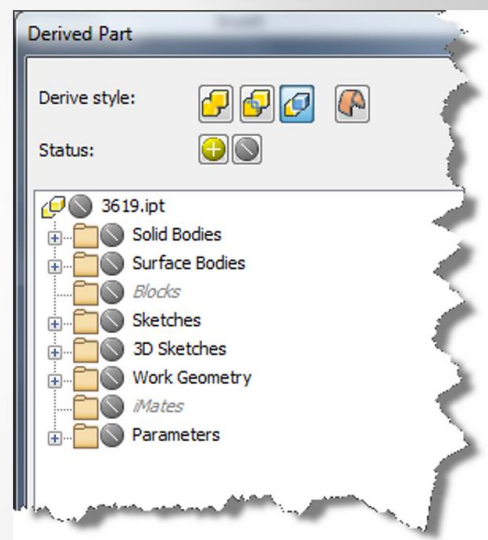
2D Sketches

3D Sketches

Work Geometry –
Planes, Axis, Points

iMates

Parameters



The Push and Pull (to and from) of Derive:

- **Pull Stuff From Selected Part File into Active Part File:**
 - **Derive Command** – Start the *Derive* Command in active part file, and pull *Stuff* in from another part
 - **Copy Part Command** – Activate the *Target Part* within the context of an Assembly (Base component must be present in Assembly as well). Start the *Copy Part* Command and select the component to copy.
- **Push Stuff from Active Part File into a New or Existing Part File:**
 - **Make Part Command** – Start command in active part file and push *Stuff* (in this case, a Solid Body or Surface Body) into a new or existing part file
 - **Make Components Command** – Start command in active part file and push *Stuff* (Solid Body or Surface Body) into a new part file (created on the fly). The Derived Part is also placed in a new or existing Assembly File
- **Push Stuff From Active Assembly File into a new Part File:**
 - **Shrinkwrap Command** – Components from an Assembly are combined, simplified and Derived into a single Part File
 - **Create Substitute Command** – Components (Parts or Sub-Assemblies) from the Active Assembly are combined, simplified and Derived into a single Part File. (the new substitute part file is also placed in Active Assembly and new level of detail is created)

Note: Derived Stuff may be pushed out from several sources, but *Stuff* only gets Derived *into* a Part File.



Using Derive in a workflow:

Let's model the Schwang, and discuss how to integrate these *Derive* functions into your workflow (or where you may already be using *Derive* in some form).

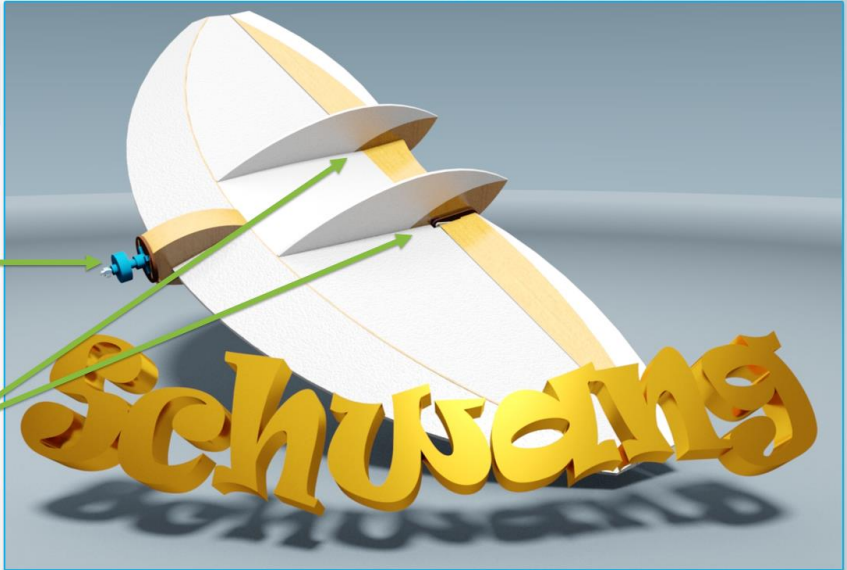
Let's Make the Schwang!

This model contains components from varied sources

Electric Motor:
Purchased Component
(measurements available, but no 3D model available)

Servos:
Purchased Components
(manufacturer supplied 3D model we can use in our design)


Remaining Parts are custom-fabricated



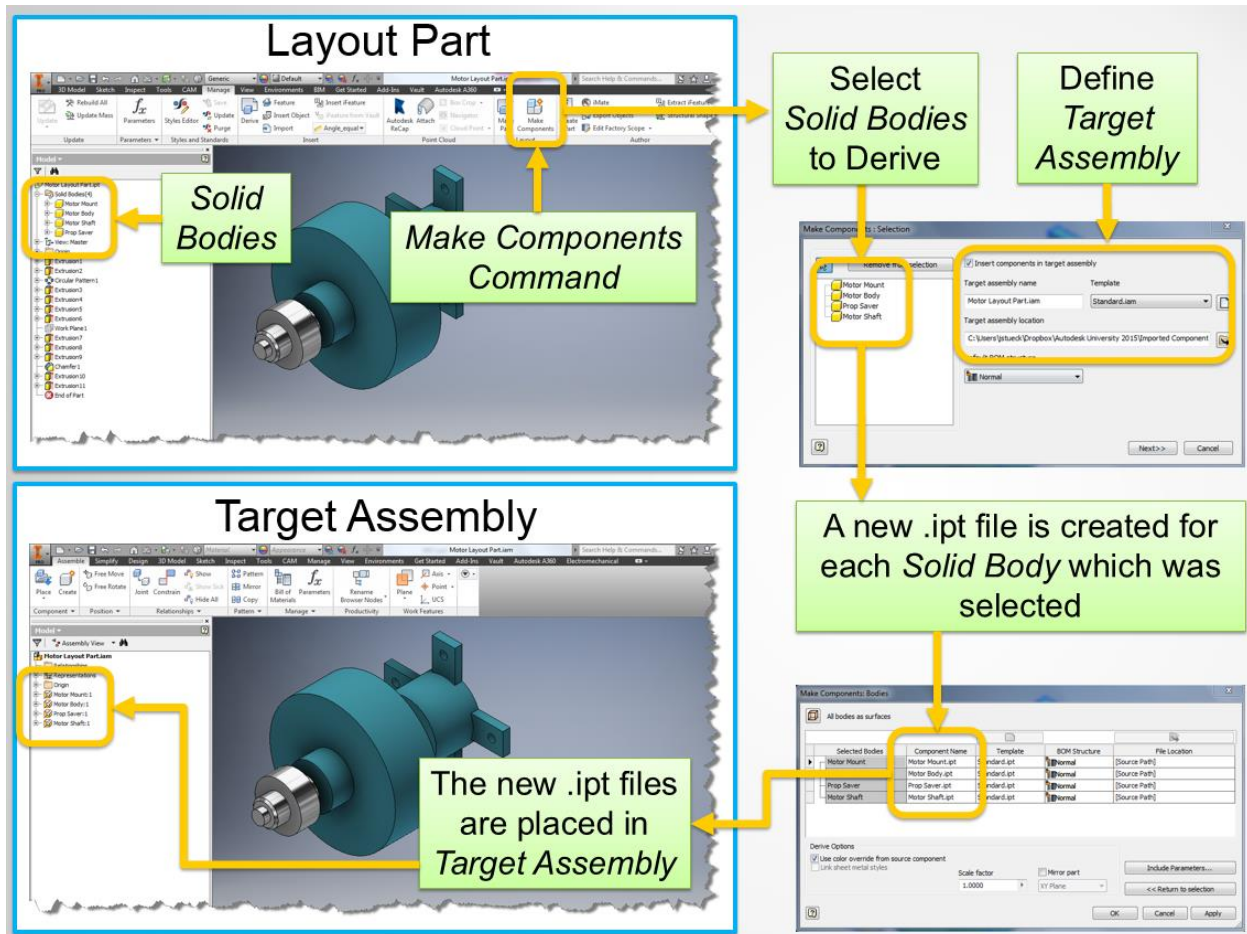
The Challenge is:

How do we build a design where the Fabricated Components are designed to fit around the size, shape, and location of the Purchased Components?

In addition, can we make the Fabricated Components?



Push Solid Bodies from *Layout Part* into *Target Assembly* using *Make Components* command:



- **So, what's the point?**

- Well, let's say the shaft diameter needs to change once you've sized all of the other components to fit around it. Now, rather than manually editing every part relative to the shaft, you can just open the Layout Part and adjust the size of the shaft. Then, all Derived instances of that shaft will update automatically.
- Not only is it faster to make changes this way, but it reduces the chances for human error being automated.

Note: The origin of each new .ipt file is aligned with that of the *Target Assembly* and then grounded in place. If the position of the *Solid Body* changes in the *Layout Part*, it will automatically update in the *Target Assembly*.



Push Assembly Components into a simplified Part using the *Shrinkwrap* command:

Active Assembly

Shrinkwrap Command

Define Target Shrinkwrap Part (Name and Location)

Create Shrinkwrap Part

New Shrinkwrap Part name: Servo Shrinkwrap
 Template: Standard.ipt
 New File Location: C:\Users\jstuck\Dropbox\Autodesk University 2015\Imported Components
 Default BOM Structure: Normal

Specify Shrinkwrap Options

Assembly Shrinkwrap Options

Style: [Icons]
 Preview: [Preview]
 Simplification: ☒ Remove geometry by visibility
 ☒ Whole parts only
 ☐ Parts and faces
 Visibility: 0 %
☒ Ignore surface features for visibility detection
☐ Remove parts by size
 Size ratio: 1 %
 Hole patching: ☒ All
☐ None
☐ Range (Perimeter) Min: 1 in Max: 999999 in
 Include other objects: ☐ Work Geometry ☐ Mates ☐ Parameters ☐ Sketches
☒ Break link ☐ Remove all internal voids
☐ Reduced Memory Mode ☐ Create independent bodies on failed Boolean

Target Shrinkwrap Part

Source listed in Model Browser (click to edit Shrinkwrap Options)

- **So, what's the point?**

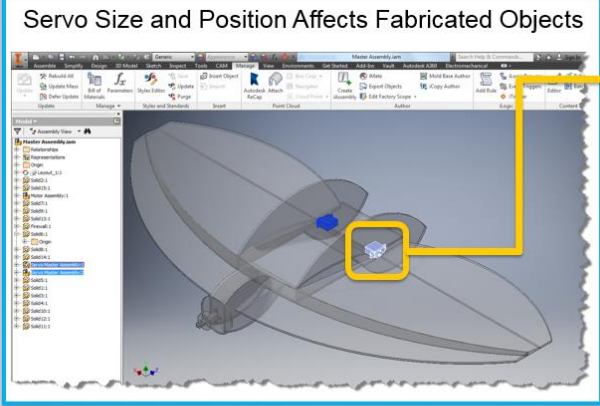
- The inner workings (the gears of this servo, for example) may not be important in the context of a larger assembly, so simplifying it will reduce the size of the file and the need for additional computing resources
- Also, you may want to share this model with users who don't need all the detail, or the full details of the assembly are classified.

Note: in addition to the Shrinkwrap command, there is also a Shrinkwrap Substitute command which will automatically place the simplified part back into the Current Assembly in a new Level of Detail.

Also, there is a Create Substitute command which allows us to simplify certain portions of the Active Assembly rather than shrinkwrapping the whole thing.



Copy objects from one part to another within the context of an Assembly:

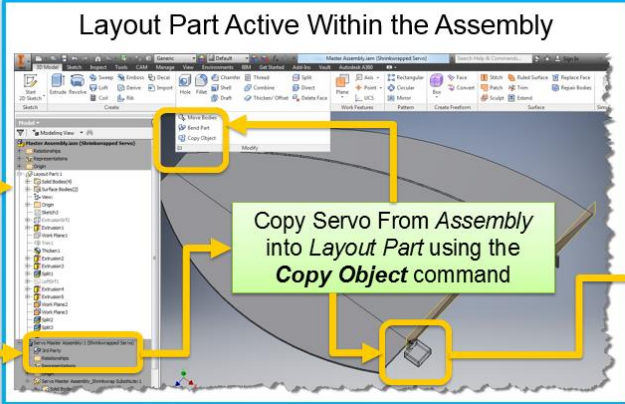


Servo Size and Position Affects Fabricated Objects

Problem: Certain objects in the layout are built around Purchased Components, so the size, shape and position of Purchased Components need to be represented in the Layout Part.

Common Solution: Measure the *Purchased Components* and manually size the fabricated components to fit (think Micrometer)

Problem with Common Solution: Time and effort of measuring Purchased Component and translating sections into fabricated components. Mistakes/inaccuracies related to the need to “envision” how the purchased components interact with fabricated components



Layout Part Active Within the Assembly

Alternative Solution: Copy the *Purchased Component (Servo)* into the *Layout Part*, and model the *Fabricated Parts* around it

Layout Part is Active within the Assembly

Since the Layout Part is active within the Assembly, The Servo Copy will be positioned in the Layout Part relative to its position to the Assembly Origin. Any changes to the Servo within the Assembly will update the Servo Copy in the *Layout Part*.

Servo has been placed into assembly and constrained into Position

Copy Servo From Assembly into Layout Part using the **Copy Object** command

- **So, what's the point?**

- If you have a micrometer at your desk, which you use to measure purchased components, or if you manually adjust fabricated components around purchased components, then this method can save time and increase accuracy.
- With a copy of the actual purchased component in your layout model, then your fabricated items can be built around the actual part, rather than measuring or envisioning how the interact.

Note: When copying an object from one part to another within the context of an assembly, make sure to make it Associative so any changes will propagate throughout all copied instances.

Also, a component can only be adaptive to one assembly. Study the reason for this to make sure that concept makes sense.



Derive Solid Bodies to create Flat Patterns in Sheet Metal Mode:

Bent Parts in Assembly

Problem: We need a flat pattern of the bent parts for fabrication purposes, but we don't want to disturb the actual Assembly Components in doing so.

Common Solution: Create a copy of each bent part using Save As. Then create a flat pattern for each one

Problem with Common Solution: If something about the bent parts change, we would be faced with repeating the process manually to update each flat pattern

Alternative Solution: Use the Derive command rather than Save As to link the flat patterns back to the bent parts so future changes will trickle down automatically

Wing Spar Derived into Sheet Metal Mode

Derived Shape

Flat Pattern using Sheet Metal tools

Derived Part

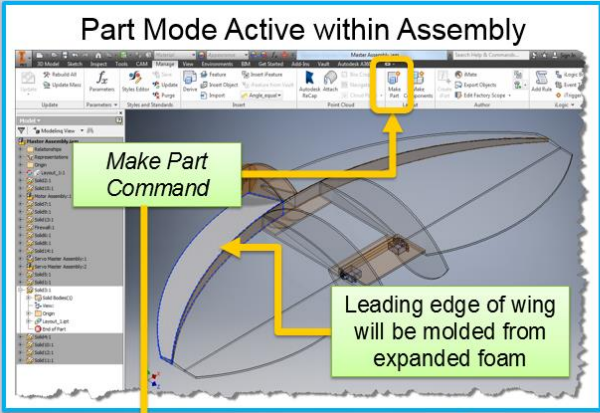
- **So, what's the point?**

- Plan for change. Even if it takes additional time to plan and utilize Derive, you'll increase your ability to make changes more quickly and accurately.
- Also, using Derive rather than Save As, will improve our ability to manage the many files which are needed throughout the product lifecycle, because there is an active link between the relevant files.

Note: The Derive Command can be used to “pull” the solid body into an active Sheet Metal Part. In a very similar way, the Make Part command can be used to “push” the solid body into a new or existing Sheet Metal Part.



Bonus Example – Create Injection Mold from Derived Bodies:



Part Mode Active within Assembly

Make Part Command

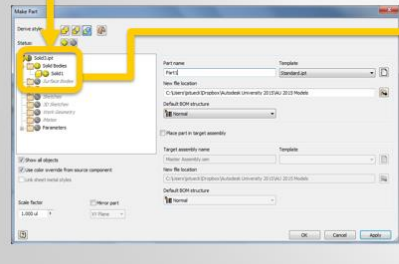
Leading edge of wing will be molded from expanded foam

Problem: We need to get the 3D shape of this part into an .ipt file so we can prepare the aluminum injection mold for fabrication.

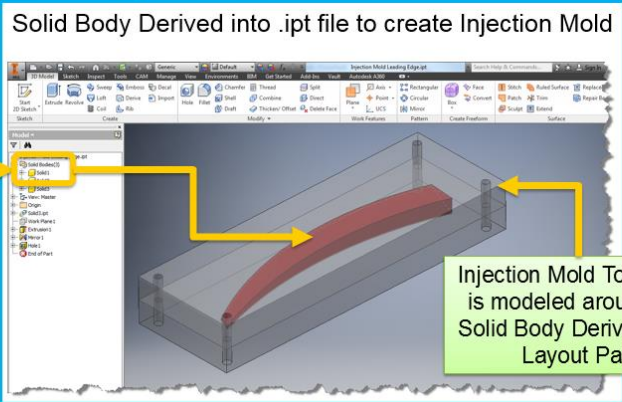
Common Solution: Save As to isolate part away from assembly, and then model the mold around the intended shape.

Problem with Common Solution: the Save As command breaks the link with the original component, so if any changes are made, the Injection Mold will need to be remodeled manually

Alternative Solution: Use the Make Part Command to Derive selected body into a new (or existing) .ipt



Derive



Solid Body Derived into .ipt file to create Injection Mold

Injection Mold Tool Body is modeled around the Solid Body Derived from Layout Part

• So, what's the point?

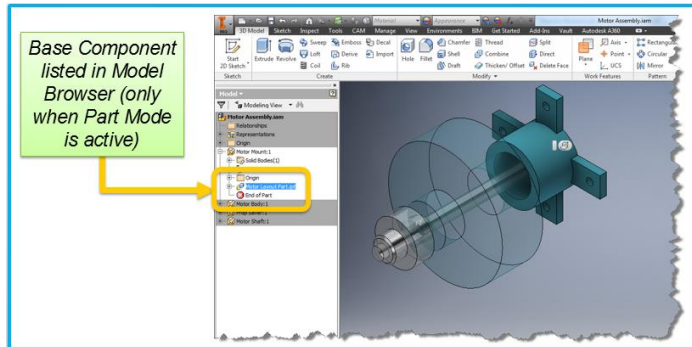
- Well, if you're at the point of making molds, then the design shouldn't be allowed to change, but we all know things like this happen. Be prepared for change.
- Also, this workflow allows you to get ahead of the product lifecycle, so you can anticipate issues before they happen. You may even discover a manufacturing limitation early enough to adjust the design before it's too late.
- If you set up your workflow this way, when it comes time to create manufacturing documents, you're already well on your way.

Note: If you're not already familiar with the Inventor Combine Command in the Part Mode, then give it a try. Even though the term "Combine" doesn't seem like the right word, you can actually perform Boolean-type operations which will add, or in this case, subtract (cut) solid mass where two bodies intersect. Thus in our case, creating a mold cavity in the exact shape of our Derived Part.

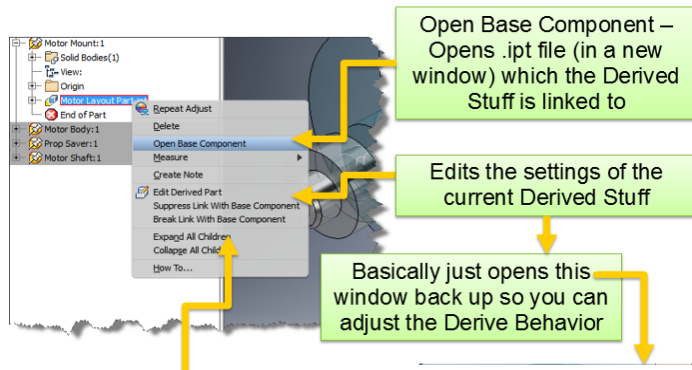


Additional Information about Derive:

The Source File of *Derived Stuff* is listed in the Model Browser of the *Target Part* File (.ipt) as a *Base Component*.



- The *Base Component* can be opened from the Model Browser and edited in a separate window (changes to the Base Component will be updated in all files into which it is currently Derived)



- The Derive settings can be edited by right clicking on the Base Component in the Model Browser and clicking *Edit Derived Part* (this just lets you change the settings of the Derived Stuff)

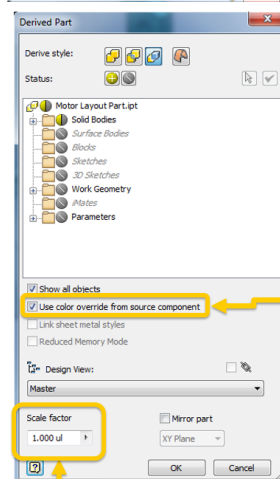
- Additional settings for Derive command are found at the bottom of the Derive Part Window such as:

- Scale Factor
- Use Color Overrides
- Mirror Part

Options also available to:

Suppress Link With Base Component

Break Link With Base Component



New to Inventor 2016 – previously, color overrides would just revert to default during a Derive operation

If you've wondered how to scale an entire model, this is the answer.



