AUTODESK UNIVERSITY 2014

Pulling Out All The Stops: Advanced Top-Down Techniques in Inventor

Walter Jaquith, Applications Expert – Imaginit Technologies

MD5792 When using Inventor software to design projects that feature complex spatial relationships, a stable and manageable workflow is critical. In this scenario, top-down design comes to the rescue by enabling the numbers to automatically drive the geometry and helping the user to avoid hours of tedious updates. In this class we will explore 2 projects in which the components are driven by performance specifications as well as geometric dimensions. In cases like these, master designers dig deep into a bag of tricks developed over years of working with top-down concepts in Inventor software. You will learn how to set up a single master file and how to structure things when a single file is just not enough. You will look at when, how, and—just as importantly—why to get a spreadsheet involved. You will also discover cool tricks for integrating purchased and stock components into a master data set. Finally, you will learn some powerful techniques for using Inventor software's automation tools to streamline your design experience.

Learning Objectives

At the end of this class, you will be able to:

- Set up master files and master data sets for top-down design projects
- Use stock and purchased components in master data sets
- Incorporate spreadsheets into the master data set
- Use automation techniques to manage and flex your designs

About the Speaker

Walt Jaquith is a Certified Inventor Professional and Certified Autodesk Instructor. A lifelong tinkerer, Walt burned through a first career as a mechanic and fabricator, and then another as a mechanical designer. After getting a preview of the Beta of Inventor R1 software, he never looked back. His current adventure involves teaching and supporting Inventor software, AutoCAD software, and Vault software as an applications expert for IMAGINiT Technologies.

wjaquith@rand.com

Introduction

In the modern design environment, the two steady constants are change and tight deadlines. The push to get products to market faster, yet reduce expenses and mistakes has driven the increased use of virtual prototyping. Given the associative nature of parametric modeling, the development of top-down processes was a natural next step. Top-down design is an inherently more accurate and comprehensive way to manage a design project. Nearly every project begins with a Design Specification or set of requirements. Often the numbers are only 'roughed in', and may not even be formalized. Regardless of what is driving the design, the challenge is to enforce the specification throughout the life and scope of the project. As the requirements evolve, how can we insure that the changes are accounted for in all aspects of the design? Top-down design embeds the specification into the geometry, insuring that all updates reach every component that they affect.

Setting up master files and master data sets for top-down design projects

	A	B	C	D
1	**NAME**	**EQUATION**	**UNITS**	**COMMENT**
2	Wheelbase	98	in	
3	Track_Front	56.5	in	Width of the Front Tires, Center to Center
4	Track_Rear	57	in	Width of the Rear Tires, Center to Center
5	Axle_Front_X	48	in	Front Axle Setback from the Vehicle Datum Point
6	Caster	3.5	deg	
7	Camber	0.4	deg	(negative)
8	CG_X	96	in	Center of Gravity X
9	CG_Z	16.5	in	Center of Gravity Z
10	RC_F	0.06	ul	Roll Center Front (factor)
11	AD_F	0.15	ul	Anti-Dive Front (factor)
12	CA_FU_L	8.5	in	Control Arm Front Upper Length

Figure 1: A complex design project begins with a Specification; a set of numbers and requirements which guide the project.

One hallmark of any top-down design project is the planning/administrative effort that is necessary at the beginning. With a conventional, top-up project, it's possible to simply start modeling parts and refine things as the project develops. In a top-down scheme, careful planning will be necessary to insure that the proper relationships are maintained throughout the project. This up-front setup time means that the design process may be well along before any models are actually produced.

This raises the question, how is the extra administrative time justified? This question must be weighed with every project; not all designs are good candidates for top-down treatment. Some considerations to account for in making the decision are listed below:

- The complexity of the geometry. If the design project consists of just a few parts, and they are mostly simple square or round shapes, then the administrative overhead associated with a top-down design scheme may not be justified. Changes can easily be made to individual parts as the design progresses. However, when the geometry itself and the individual parts' relationship to each other are more complex, a change to a single dimension can result in hours reworking the models.

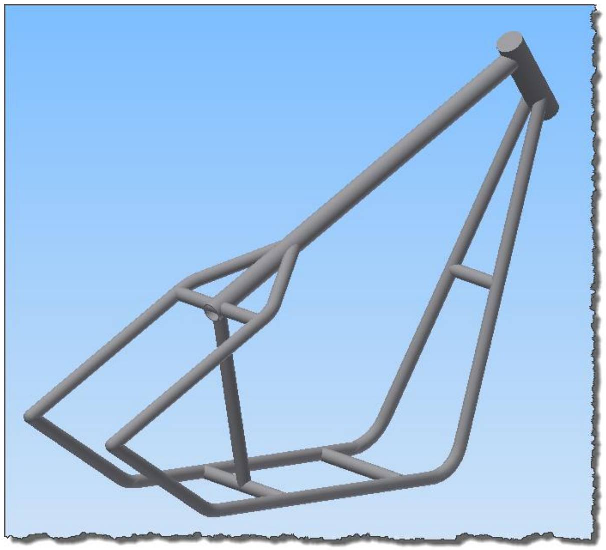


Figure 2: Tube frame projects are often more complex than they appear due to the unusual angles and interactions between the individual tubes.

- The number of expected design changes. Some designs are fairly static by their nature. If the design can be expected to go from concept to production with few changes, then the time spent setting up a top-down design scheme might not be justified. If the design's final configuration is unknown at the outset, and there are many factors expected to develop which would affect the design's final form, then spending the time to plan for those changes from the beginning of the design process makes good sense.

- The nature of the expected design changes. Many design projects can be readily defined by a series of driving parameters which not only define the geometry, but also indicate the ways the geometry could be expected to change. These projects are good candidates for top-down treatment. If the design is prone to change in ways which would introduce an entirely new set of driving parameters, then top-down design would be a less efficient approach. A change of that nature might require starting over from scratch. Our sports car suspension project easily illustrates this idea. The center of gravity (C/G) is a foundational parameter for that design. While the location of the chassis' C/G might change significantly, it is extremely unlikely that the C/G as a parameter would become unimportant and be replaced by some other, completely different parameter. Since most of the driving parameters for that project are of that type, it is a good candidate for top-down design techniques.

- <u>How many versions of the design will be needed</u>? While a single version design might still be a good top-down project, the need to produce multiple versions of the design makes it much more likely. This follows the same logic which justifies using iParts to create families of components. Once the first version is established, copying the entire design and plugging in a new set of numbers will quickly and easily produce the desired variations.

If the project meets the criteria above, then the benefits of top-down treatment should easily repay the up-front time and effort it requires.

Design Parameters

The first step in the process is to identify the critical, <u>Driving Parameters</u> for the project. These are often dimensional numbers, but might also be weight or capacity requirements, performance specifications, or footprint and envelope conditions. If any existing or off-the-shelf components are pivotal to the design, these should be documented in the spec as well.

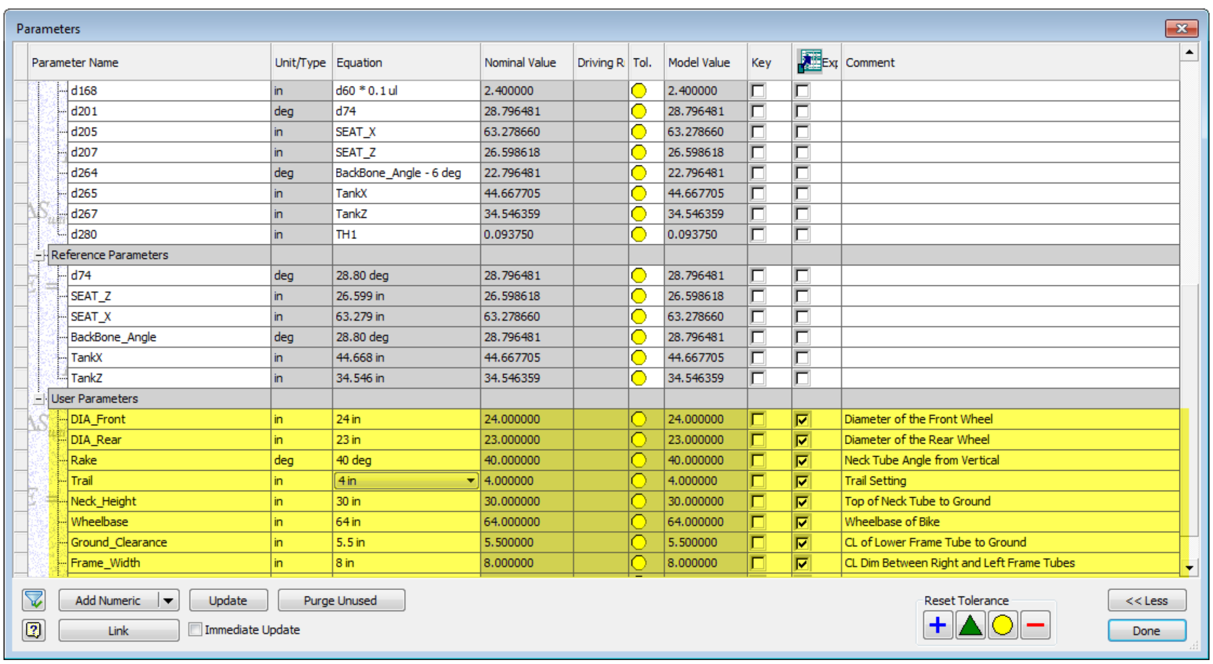


Figure 3: The parameters Dialog box from the chopper frame master file shows the driving parameters (highlighted), and some of the places they are used in configuring the rest of the model.

As a general rule, the number of driving parameters should be kept to the minimum required. Adding parameters to the list which do not directly drive the geometry in some way will add needless complexity and confusion. Later in the lesson we will look at what to do if some elements of the spec need to be calculated from the master geometry. In that case, iLogic tools can be used to write the parameter back to the master spreadsheet.

An important factor in setting up the dataset is allowing for the unknowns, estimations and flat-out guesses which often attend the beginning of a project. If the project is fairly simple, then traditional modeling methods will serve. Any changes will involve updating each part to reflect the new data, and then adjusting the assemblies as needed. For more complex projects, some form of centralized management will be needed to keep on top of the design.

As with any design project, the challenge is to translate the specification into a compliant, functional machine. With a top-down scheme, the spec will first need to be put into a form which can actively drive the Inventor models.

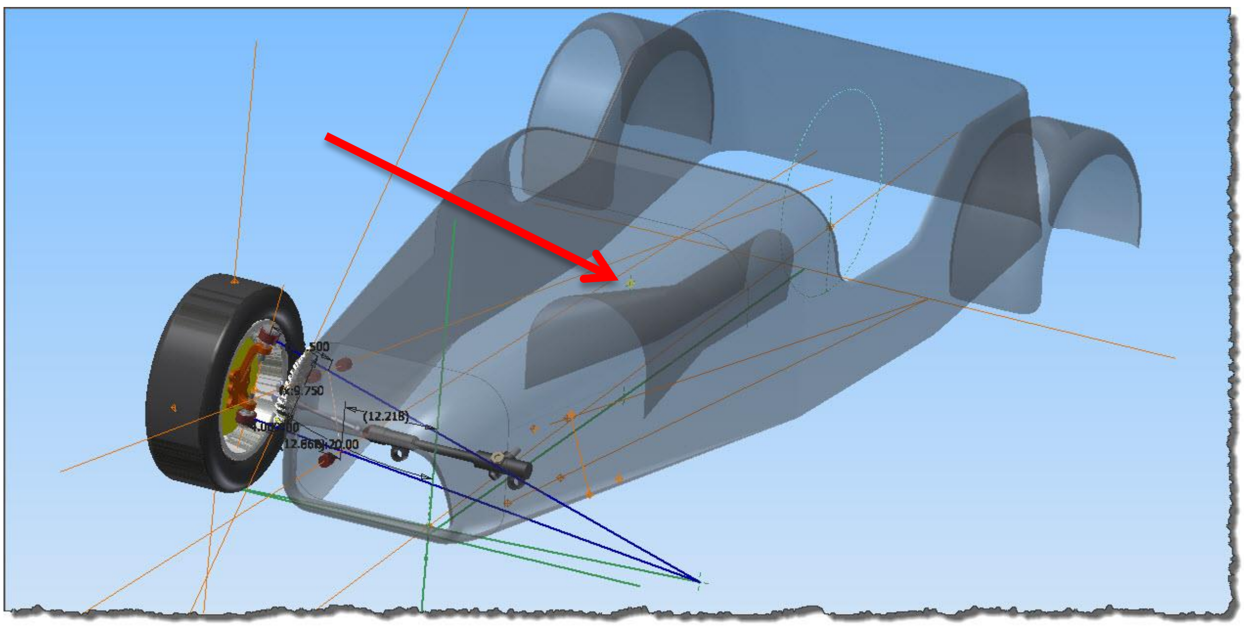


Figure 4: In the S7 Sports car, much of the suspension geometry is configured based on the vehicle's center of gravity, a point which can usually only be estimated early in the design cycle.

Inventor Master Files

A good strategy for managing the design specification is to group the critical elements of the project in one container where they can be easily accessed. This is elemental to any top-down design strategy. Rather than spreading the critical parameters through the dataset by embedding them in various part files, they are instead organized and placed in a <u>Master File</u>. This course examines two kinds of master files: Inventor component files, and Excel spreadsheets.

For less extensive projects, the Parameters Dialog in a part file (Figure 3) is a good place to contain driving parameters. Some of the controlling data for a project may be graphical in nature, and so the same part file might also contain important sketches, work geometry or even regular 3D geometry which defines the project. In a case like this, the Inventor part file becomes the master file for the project. In some of the simplest cases, the master file can also serve as one of the component files in the design assembly. While this can form a very compact top-down project, care must be taken to document the project adequately so that other Inventor users who open the files can easily understand how the files are structured.

In many advanced cases, the need to incorporate purchased components and other existing geometry into the master dataset creates the need for assembly files to be used. For these projects, there will not be a single master file, but a set of masters comprised of associated part and assembly files. Pulling the driving parameters out into in Excel spreadsheet becomes an excellent way to organize the project. The spreadsheet can be linked into each of the master files, creating a direct relationship that is simple and robust.

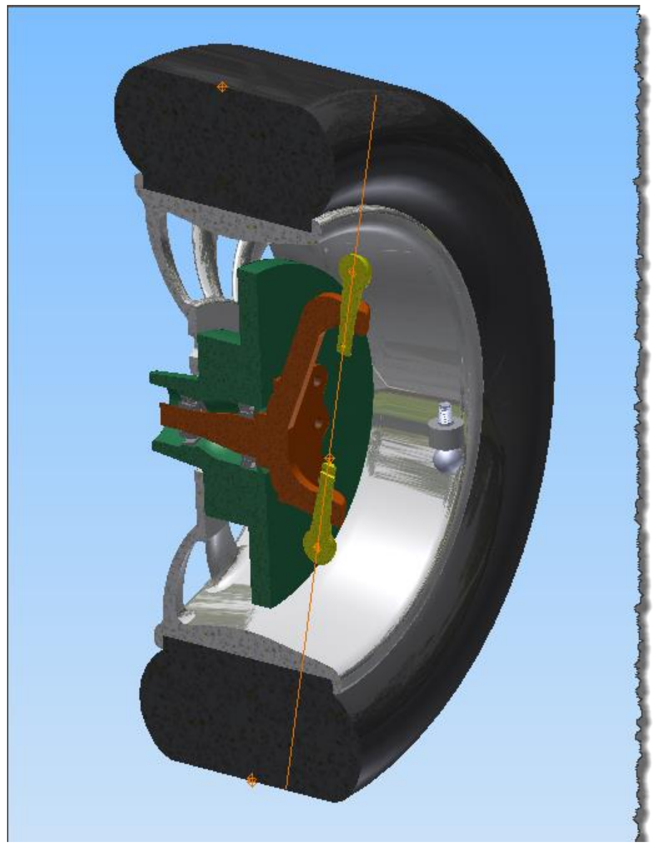


Figure 5: To accurately place the ball joints, we need the entire tire-spindle assembly in addition to the numbers such as toe and camber.

In determining how a project's master dataset should be configured, it is helpful to consider what we want to get out of it. The point of a top-down data scheme is to be able to anticipate and even explore logical changes and variations in the design before committing to production documents. In most cases the final design will be decided based at least partially on feedback from the design process. In some cases the viability of the design itself may be contingent on testing to be conducted on the master dataset. Accounting for any and all testing needs is an important element of the structure of the master dataset. In any case, a master dataset should allow you to explore the design—whatever that means for your particular case—easily and painlessly.

Example 1: Hardtail Chopper Frame

Our first example is a simple motorcycle frame…or is it simple? Although the frame doesn't have many parts, and isn't highly technical or exceptionally complicated, as a design project it does have several elements which make it a good candidate for a top-down approach:

- External Driving Parameters. Many of the numbers which ultimately determine the layout and dimensions of the frame are not on the frame itself. They are things like tire diameters, wheelbase and the configuration of the front fork assembly. When those elements change, the frame must change to accommodate them. This is typical of any type of frame structure; it ultimately exists to mount and support the parts of the machine which actually do the work. It makes little sense to begin work on the frame before we've defined the critical elements and the ranges in which they might vary. Armed with those numbers, we can easily and quickly design a frame that fits them.

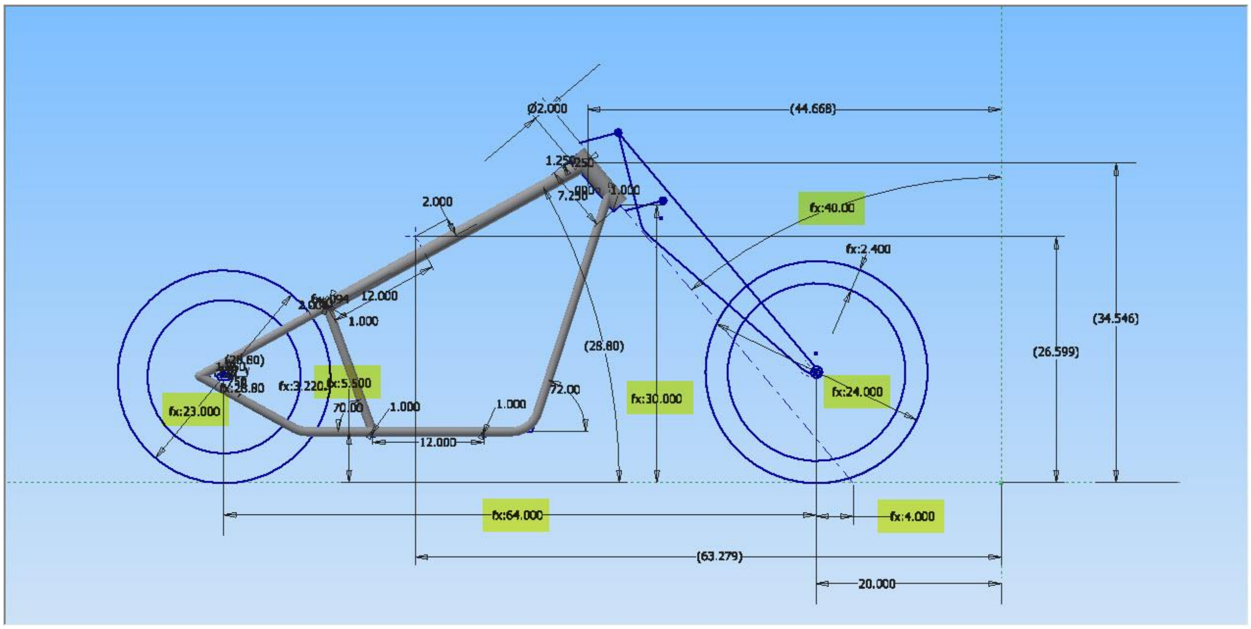


Figure 6: For the chopper frame, many of the driving parameters (highlighted) are not actually frame dimensions. The handling characteristics, front suspension configuration and tire size drive the frame.

- Integration of the individual parts. No matter how simple in concept, there is at least one aspect of a motorcycle frame that is very complex: The relationship and interaction of its individual tubes. Because of the way the individual frame members fit together, any change to one of the master parameters is likely to change every tube in some way. The copes and bend angles are not square, perpendicular or aligned to any origin axis. Instead they are aligned to critical mounting points for the bike's major components, and a simple wheelbase tweak can change it all. Using conventional modeling methods, this would entail tedious hours of updating individual part files, making sure everything aligned and matched. With a well-designed top-down project, changing the driving parameter itself will update all the models together, insuring that no critical relationship is lost in the update.

The chopper frame begins with 8 driving parameters which control the configuration. Because the majority of the layout of the frame is 2-dimensional, all but one parameter (the one which controls the frame width) can be incorporated into a single profile sketch. A second sketch placed along the backbone will consume the frame width parameter. The 3D paths for the tubing runs will be associated to the two main sketches, and the tubes will be created as individual solids for future export to a weldment assembly.

Using Stock and Purchased Components in a Master Dataset

As far as top-down projects go, there's nothing too tricky about the chopper frame at this point…but that's about to change. Many of the design considerations for a custom chopper largely revolve around aesthetics and fitting off-the shelf components into a non-standard project. In order to use the master file to check these elements, we need to bring some purchased components into the picture. In this case, one of the major considerations of both function and aesthetics is the relationship between the bike's seat and fuel tank.

Bringing in Geometry

The two required parts can be brought into the master file using the derived part tool. The process is straightforward, but the geometry will come in aligned according to the file's origins, and will need to be moved into position.   With careful management, they can be positioned in a manner which will keep the geometry in place when the frame's configuration changes.

To accomplish this, we need to add some reference geometry to the master file which will define where the origins of the tank and seat need to be (Figure 7). Once those numbers are established, the <u>Move Bodies</u> command can be used to move the geometry into place.

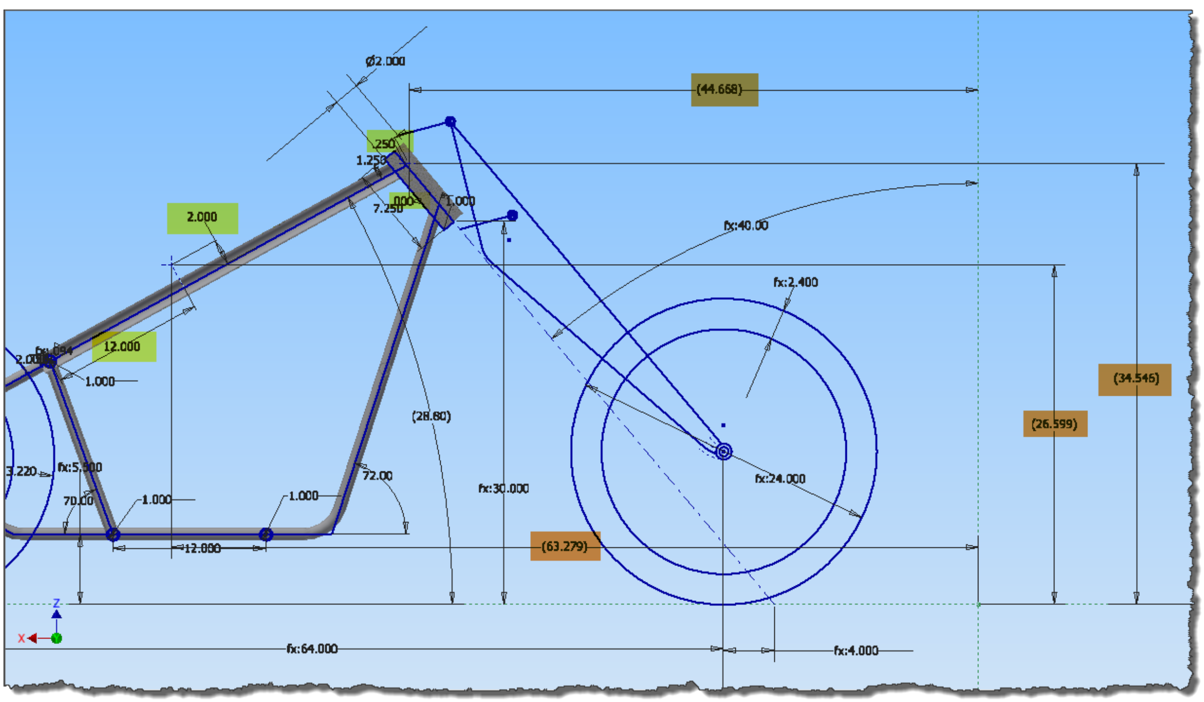


Figure 7: Points for placing the seat and tank are dimensioned off the frame, but reference dimensions are added to the file's origin point. These driven parameters are used to move the seat and tank into position after they have been derived in.

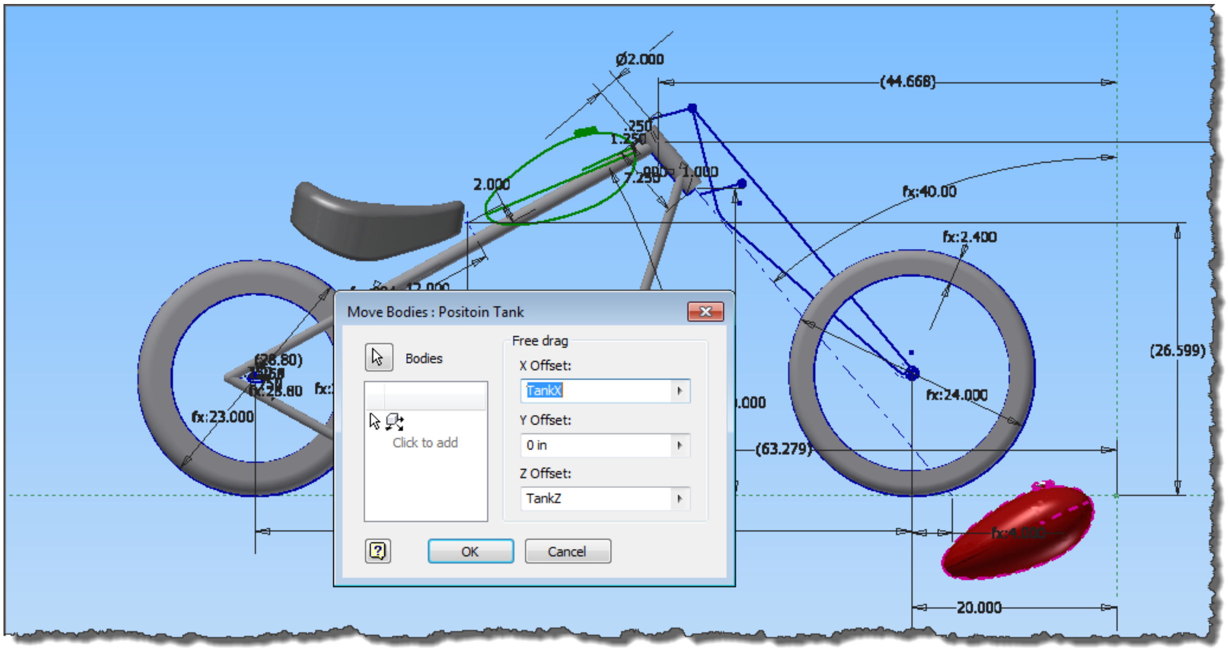


Figure 8: The reference numbers are used to position the tank, insuring that it moves correctly with the frame when the design changes.

This method of incorporating stock and purchased components into a top-down master dataset does have some limitations:

- There is a considerable amount of planning and setup needed for each component which is brought into a master file in this manner. The complexity of the process makes it practical mainly for smaller projects. Since more complex top-down projects often move away from a single master part file for other reasons, this does not detract too much from the technique's usefulness.
- The Move Bodies command moves only solids; it does not move any derived work geometry, origin geometry, or sketches which might otherwise be useful. Those will stay at the origin point when the solid geometry is moved. This is as designed; the tool's purpose is to move solids. In this case, however, it means that the technique is less useful than it might be.

We'll take another look at working with stock and purchased components a bit later…

Incorporating Spreadsheets into a Master Dataset

The simplicity of using a part file to contain driving parameters makes that workflow well suited for less extensive projects. When things get more ambitious, the advantages of separating the important numbers from the geometry begin to multiply. In these cases, the high level of integration between Microsoft Excel and Inventor make Excel a natural candidate to contain and manage the raw numbers which drive a top-down project. Some advantages of using a master spreadsheet file:

- Management-level administration. A spreadsheet file can be administrated by a manager who need not be an Inventor user. The spreadsheet file can also be used to communicate the design parameters to non-Inventor users.
- Simplified file structure. Excel files can be linked directly to any Inventor component file. This means the master spreadsheet can be tied directly to each of the component master files in a complex project. This simplifies the matrix of relationships between the files, making the whole project robust and manageable.
- Automation options. Excel's iLogic integration means that parameter entries can be manipulated programmatically, and made subject to rules and conditions.

To format a spreadsheet for use in an Inventor design project, four columns are required. The data can begin anywhere on the first sheet (See Figure 1).

- Name. Corresponds to the Parameter Name column in the Parameters Dialog. The normal restrictions on parameter names apply.
- Equation. Corresponds to the Equation Column.
- Units. Corresponds to the Unit/Type Column.
- Comment. Corresponds to the Comment column. Entries in this column are optional.

Spreadsheets can be linked into the Inventor files using the tools in the Parameters Dialog. Once the linking operation is initiated, it is important to specify the Start Cell, where the first parameter name is found. In our example, the start cell is "A2" (Figure 9). Inventor will begin reading at that cell, and link the entries reading down in a table 4 columns wide. Any number of parameters can be imported in this manner, and the procedure is not restricted to a single spreadsheet file.

The linked parameters are placed in the Parameters Dialog Box, but cannot be changed in Inventor; they must be updated in the spreadsheet itself. They can be checked as key parameters or tagged for export if needed. Linked spreadsheets appear in the 3rd Party folder in the Browser. From there they can be opened for editing, deleted, or the source file can be changed.

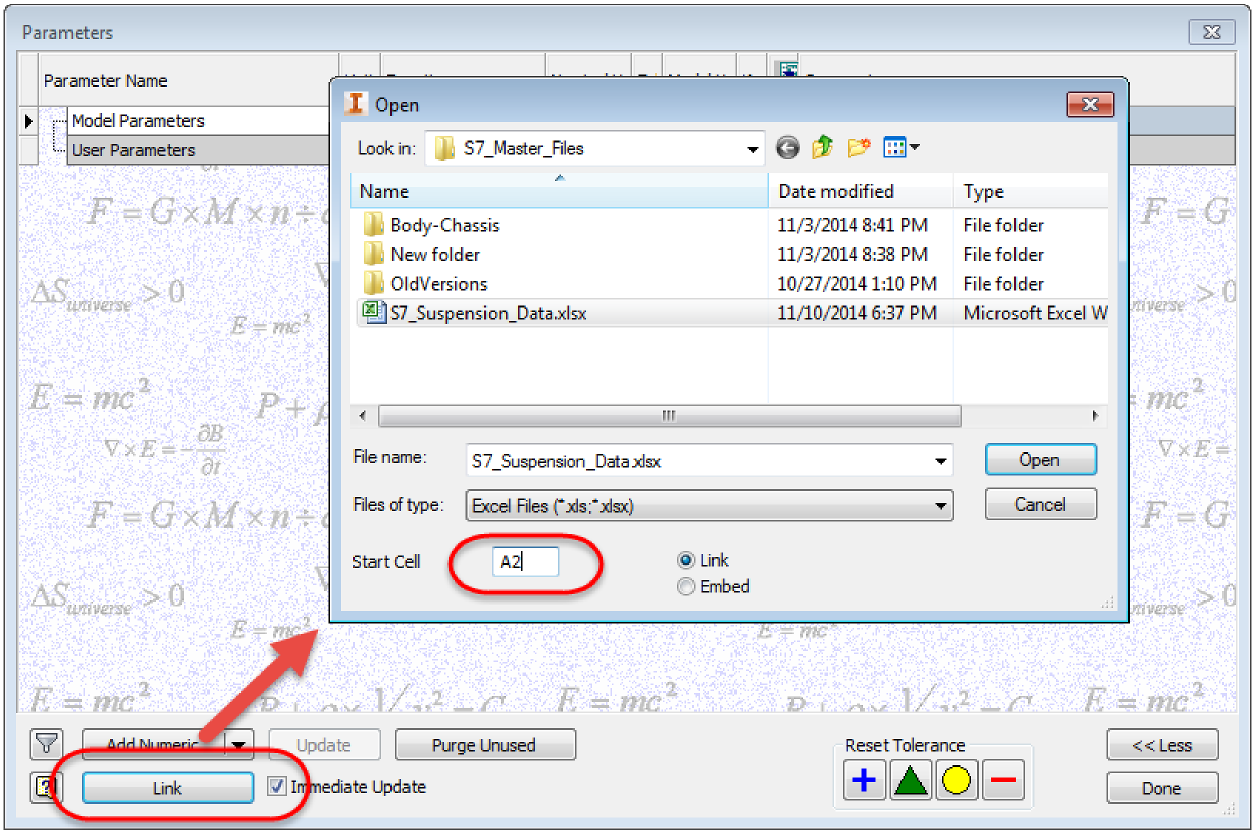


Figure 9: Spreadsheets can be linked into Inventor component files from the parameters dialog box. It's important to specify the start cell.

Example 2: Independent Front Suspension

Where the chopper frame's complexity was more subtle, there is nothing straight or easy about our second example. The geometry is off-axis on all three dimensions, and is driven by a large number of diverse parameters. Few elements of the design are straight, perpendicular, or defined by simple, direct dimensions. Instead of being introduced later in the process, purchased components come into play at the outset, and there are entire subassemblies which must be placed before any fabricated parts can be defined. Finally, the geometry of the entire project is based on numbers—such as the vehicle's center of gravity—which will likely have to be estimated at the outset. This is a project which would be very difficult to complete using 'conventional' bottom-up modeling techniques.

In a case like this, a single master part file will be inadequate, and the sheer number of driving parameters will make managing them inside Inventor unwieldy. Two characteristics of this project will be the Excel spreadsheet used to manage the driving parameters, and the multi-tiered master set of Inventor part and assembly files.

Front Suspension Master File Structure

The method of developing the master file structure of the front suspension project is fairly straightforward, even if the result is complex. Start with a part file, and proceed until the data from stock or purchased components is needed in order to advance. In this case, the method from the earlier example—bringing derived geometry into the master part file—will be unwieldy. It will be much better to move to an assembly, bring in the first master part and then introduce the needed components. Eventually, the time will come when more sketch geometry is needed. The assembly will then be derived into a new master part file, and the process will begin again. With each file, the master spreadsheet is linked in directly, giving access to all the driving parameters.

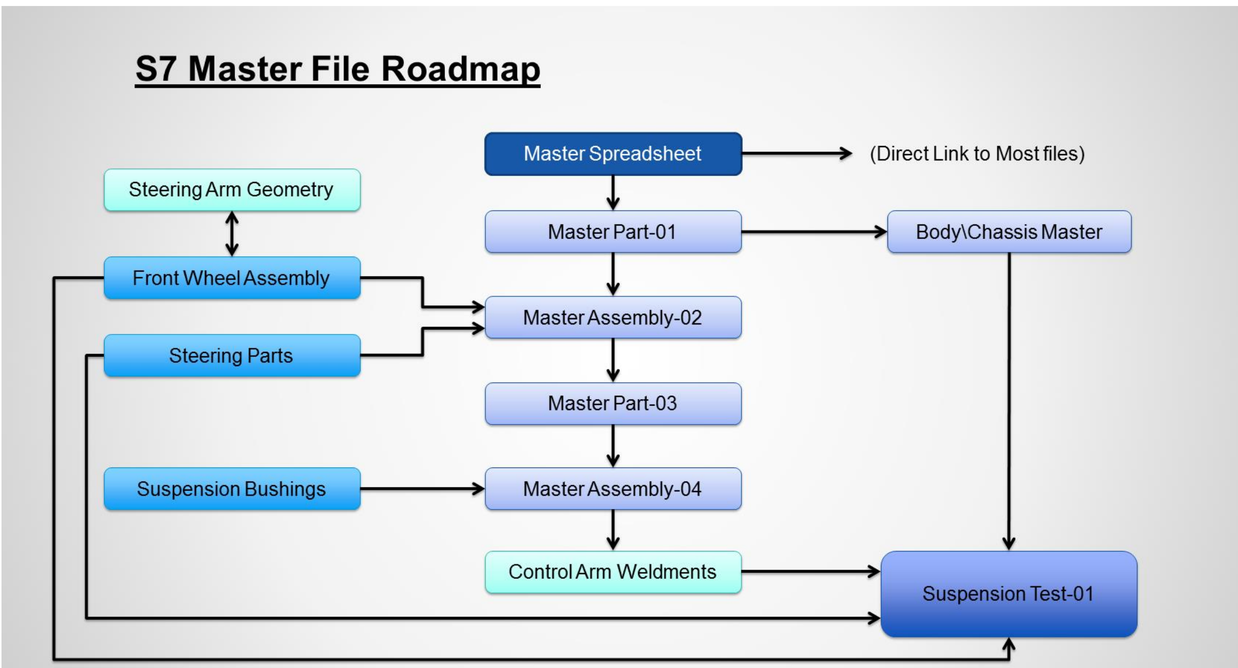


Figure 10: For a project as advanced as a sports car's suspension, the master file matrix will necessarily be fairly complex.

Just as any number of sub-assemblies can be incorporated into the master dataset, the master files themselves can be used multiple times, to 'spin off' elements of the design as needed. In our current example, In addition to forming the foundation of the rest of the master dataset for the suspension, the first master part file is also derived into a file which forms the basis for the car's body. It will also likely be used as the basis for the rear suspension's master dataset. In this way, all the various design elements of the project are linked, and a change to the driving parameters in the spreadsheet will update them all.

Through the course of setting up the master dataset, stock parts can be introduced, and fabricated components defined as needed. In certain cases, only critical elements of the components need to be defined early in the process. Doing too much too early can lead to needless tweaking as the design evolves. In our S7 Suspension example, it is important to establish the location of the outer steering arm pivot point early in the design process, so that part is defined in the front wheel assembly. There is no need to complete the design; that can and should be done after the entire configuration is much more settled. Only the ball joint mount itself is needed to verify the placement of the steering rack.

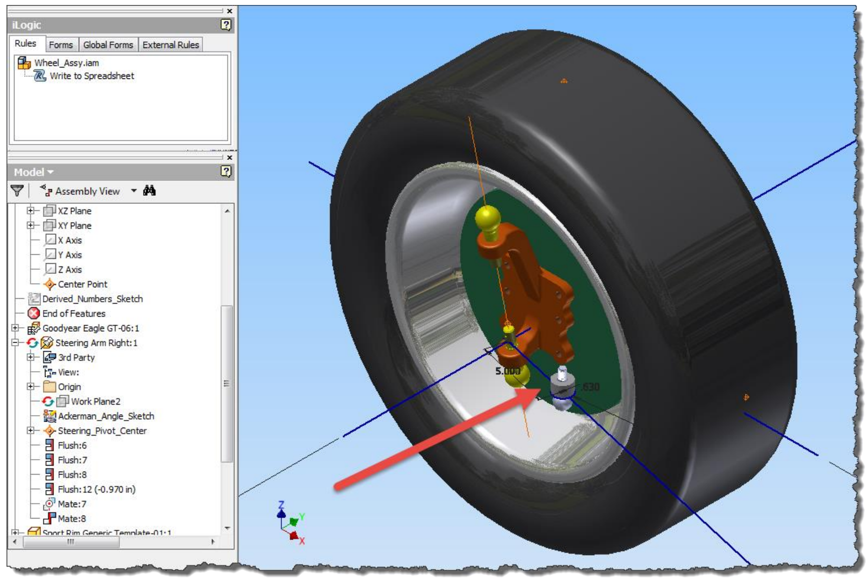


Figure 11: The steering arm doesn't need to be completely designed until later. Only the mounting boss for the tie rod end is required at this point.

Use automation techniques to manage and flex your designs

iLogic is a powerful, Visual Basic-based programming environment which can be used to manipulate Inventor files in a nearly endless number of ways. As with any programming language, mastering the subtleties of iLogic takes time and diligence. However, many of iLogic's most useful functions are quite simple to implement. Given that the whole point of a top-down design approach is to automate the design process, augmenting a master dataset with iLogic functionality is a natural step. For the purposes of this class, we will look at three specific cases where iLogic can assist us in managing top-down design projects.

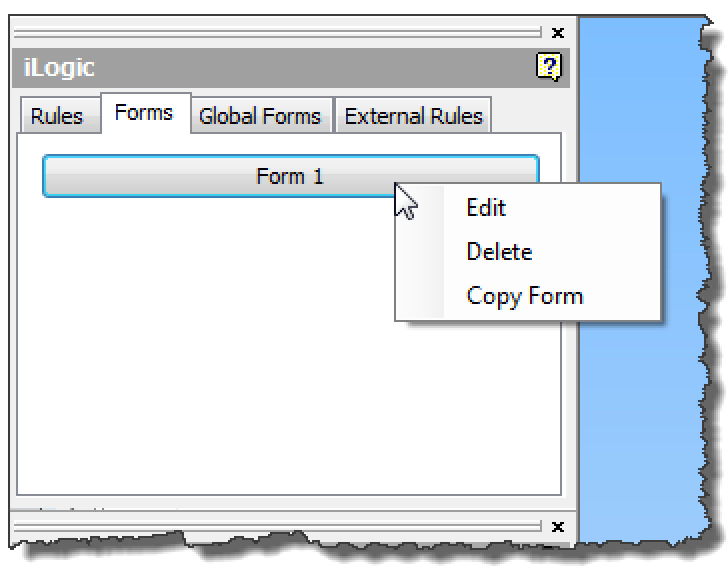


Figure 12: Forms are created and edited on the forms tab of the iLogic Browser.

iLogic Forms and Sliders

iLogic's ability to create forms is a powerful tool in any designer's toolbox. One advantage of this tool is that it requires no coding at all, and can be used without creating a single iLogic rule. Any parameter can be manipulated by placing a control in an iLogic form. There a several different types of controls available, depending on the type of parameter: Regular parameters can be either text boxes or sliders. Multi-value parameters can use a combo box, list box or radio group.

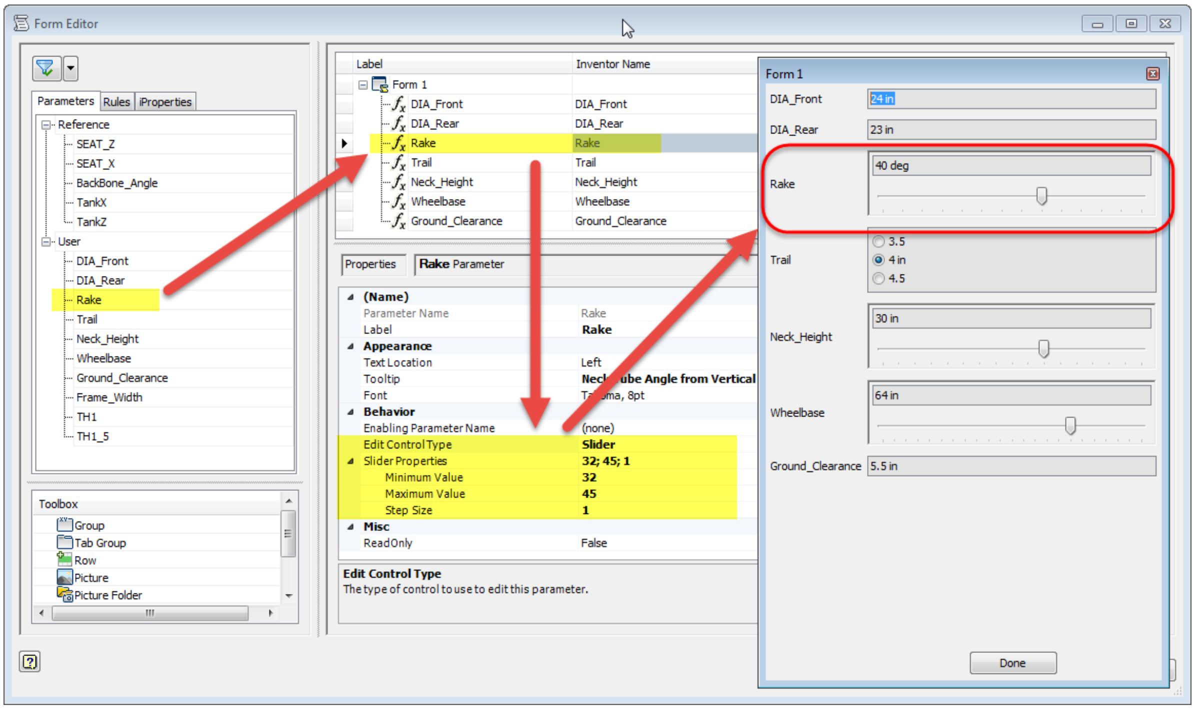
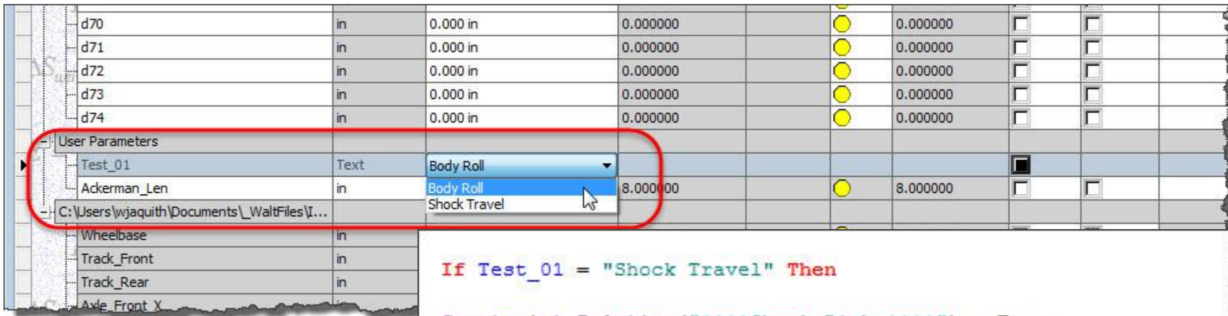


Figure 13: To add a parameter to a form, first drag it from the parameters tab, then select the control type.

Conditional Configuration for Testing

In certain cases, the testing requirements may involve changing the status of several assembly constraints. In our test assembly, either the suspension travel or the body roll can be flexed, but the two require the geometry to be constrained in different ways. This involves suppressing and unsuppressing a series of assembly constraints depending on which test is being run. For the body roll test, the constraints representing the shock absorbers must be suppressed, and the tires pinned down to the ground plane. To test suspension travel, those conditions must be reversed. To make things easier a rule is set up to make the changes based on the state of a user parameter. Toggling the user parameter will trigger the rule, which will then make the




```
If Test_01 = "Shock Travel" Then

Constraint.IsActive("****Shock Right****") = True
Constraint.IsActive("****Shock Left****") = True
Constraint.IsActive("****Pin Tire Left****") = False
Constraint.IsActive("****Pin Tire Right****") = False

End If

If Test_01 = "Body Roll" Then

Constraint.IsActive("****Shock Right****") = False
Constraint.IsActive("****Shock Left****") = False
Constraint.IsActive("****Pin Tire Left****") = True
Constraint.IsActive("****Pin Tire Right****") = True

End If
```

Figure 14: This rule manipulates 4 assembly constraints in the test assembly based on the value of a parameter called "Test_01".

required changes to the constraints in a single operation. In this way, a single action from the user can make any number of changes to the model. This is just one of the things at which iLogic excels.

Write derived values to the Master Spreadsheet

Some parameters which should be included in the master spreadsheet cannot be solved until the design process is well underway. For example, in the S7 suspension, the length of the tie-rods cannot be known until the first master assembly, which places the inner and outer pivot points, is complete. This parameter will be used in the test assembly to position the steering arm, so the master spreadsheet is a natural place to keep this information. iLogic tools will be used first to measure the distance between the inner and outer pivot points for the steering, and then write that number to the master spreadsheet.

Note: The master spreadsheet needs to be closed when this rule runs. If it is open in Excel this rule will return an error, as the file is considered to be locked by the user which opened it.

```
'This rule measures the tie-Rod length from the inner pivot To the ball-Joint
'And writes the value To the master spreadsheet.

TRL = Measure.MinimumDistance("Rack_Pivot_Point_Right_Inner", "Rack_Pivot_Point_Right_Outer")

ExcelFile = "3rd Party:S7_Suspension_Data.xlsx"

GoExcel.CellValue(ExcelFile, "Parameter_Link", "Tie_Rod_Length") = TRL

GoExcel.Save
```

Figure 15: A simple rule is used to write calculated parameter values back to the master spreadsheet.