Roger Toogood, Ph.D.

Creo® Parametric 9.0 Advanced Tutorial
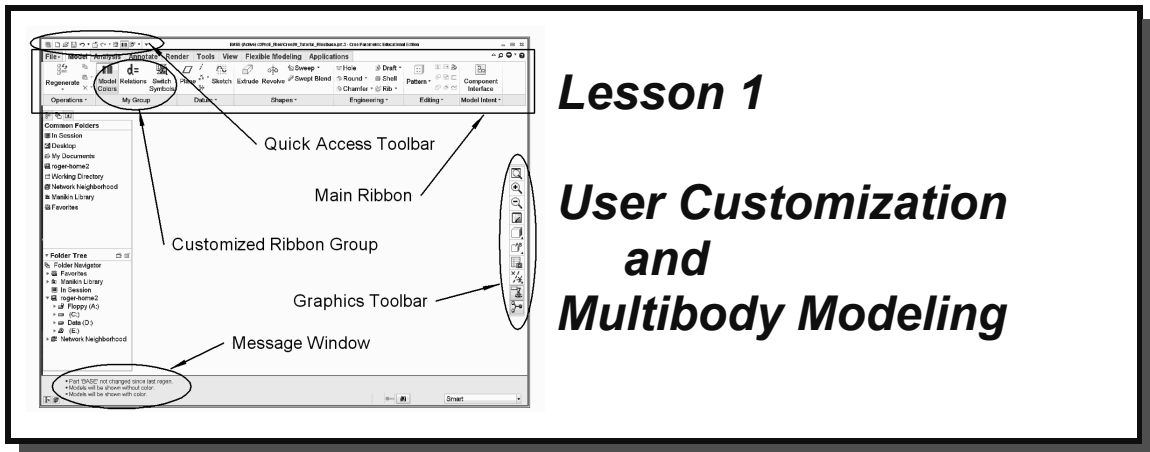
Visit the following websites to learn more about this book:

SDC PUBLICATIONS

amazon.com

Google books

BARNES & NOBLE

Lesson 1

*User Customization
and
Multibody Modeling*

Synopsis:

Configuration settings; customizing the screen toolbars and menus; mapkeys; part templates; introduction to multibody modeling; introduction to the project

Overview

This lesson will introduce tools for customizing your working environment and show you how to create some useful shortcuts for accessing Creo commands. The intent of these tools is to let you set up the program to suit your own preferences so that you can work most comfortably and efficiently. We'll also see how you can create your own part templates. In the second half of the lesson, the basic operations of multibody modeling will be used to create a simple assembly using a master model. Finally, the major project used in this tutorial is introduced and the first four parts are presented.

User Customization

Customization principally involves two categories as follows.

Configuration Options (*config.pro*) include such items as
- environment options (decimal places, units, location of resources)
- model and entity display options
- assembly and detailing options
- Sketcher options
- and much more (close to 1000 options!)

Changes to these options generally apply to the current session only, unless explicitly saved. Some options will require a restart of the program.

Interface Customization Options (*creo_parametric_customization.ui*) deal with
- system colors

- •   configuration of command buttons on ribbons
- •   contents of Quick Access toolbars
- •   selection options
- •   and other aspects of the user interface

Changes made to these options are saved automatically. These configuration/interface option files are loaded automatically at startup. Let's get started.

Configuration Settings

Launch Creo, or if it is already up erase everything currently in session and set your working directory to your normal start-up directory. You should now be looking at the **Home** ribbon, with nothing loaded.

Figure 1 The **Settings** group in the **Home** ribbon

The **Settings** group in the **Home** ribbon is shown in Figure 1. This group is available only in the **Home** ribbon. Select either of these commands (*Model Display*, *System Appearance*), or the arrow in the lower right corner, and you will see that each is going directly to one of the option areas in the same **Creo Parametric Options** window. At all other times (i.e. when you have a part or assembly loaded), this window is available by selecting *File ➤ Options*.
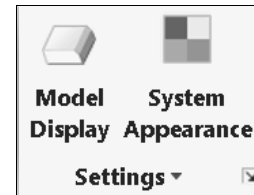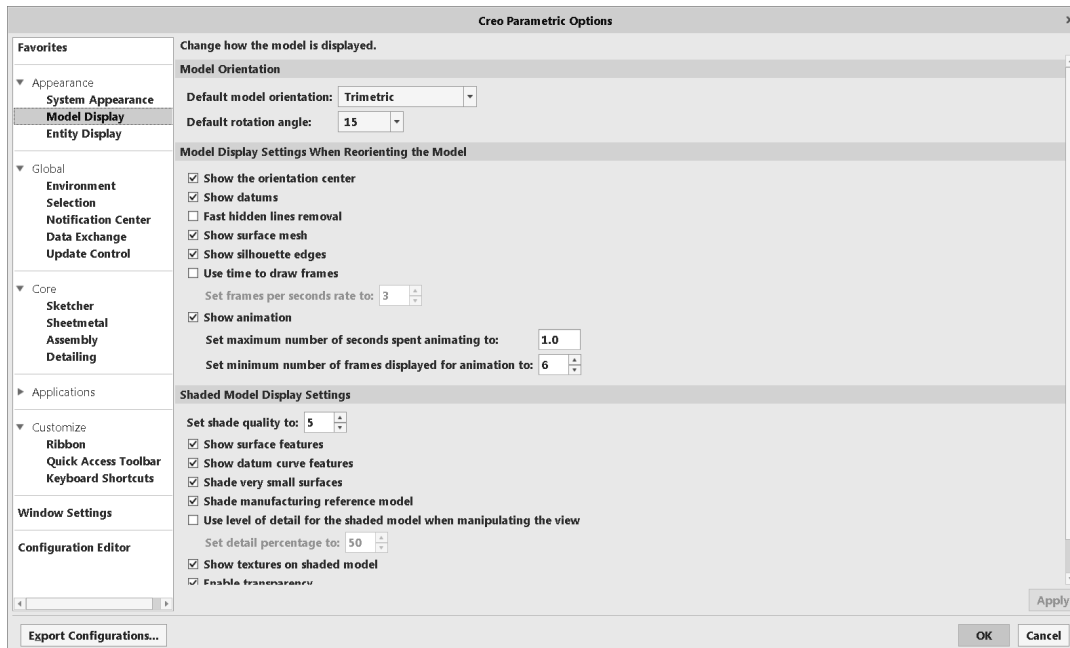

Figure 2 The **Model Display** panel in the **Creo Parametric Options** window

The **Options** window is the main location for most of the customization tools. Select the **Model Display** category as in Figure 2. The panel on the right shows all the settings related to the chosen category. Have a look at some of the other categories (like Sketcher) and options available.

Any setting changes made in this window (for example: Sketcher decimal places) will be valid only for the current session unless you explicitly save them. That is done using the **Export Configurations** button at the bottom. This opens the window shown in Figure 3. This will likely show the **Common Files/Text** area of your program installation, with several *.pro files listed if you scroll all the way to the right. The default configuration which is loaded at startup is *config.pro*. We will leave that one alone, so select **Cancel**. We will discuss saving our own copy of *config.pro* a bit later. In addition, at the bottom of some of the panels in the Options window will be an **Import/Export** button (that also allows you to read in a file of previously created settings).
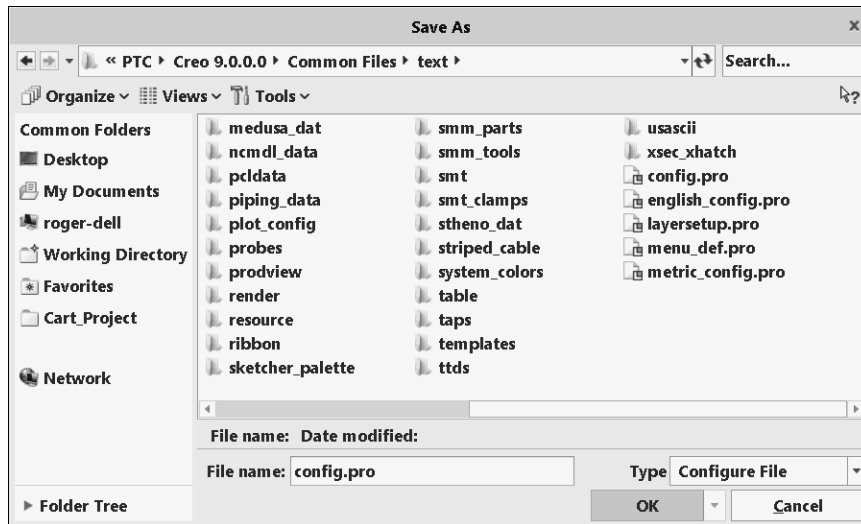
Figure 3 Saving the configuration settings

Now let's have a closer look at the contents of the *config.pro* file itself.

Configuration Files (*config.pro*)

As mentioned above, the configuration file *config.pro* is automatically read when you launch a new session. You can also read in (and/or change) additional configuration settings at any time during a session. For example, you may want to have one group of settings for one project you are working on, and another group for a different project that you switch to during a single session. In this tutorial, we will deal only with the use of the single configuration file, *config.pro*, loaded at start-up.

Several copies of *config.pro* might exist on your system, and they are read in the following order when Creo is launched:

♦ *config.sup* - this is a protected system file which is read by all networked users but is not usually available for modification by users. Your system administrator has control of this file.
♦ The CommonFiles/Text directory - the *config.pro* file stored here is read by all users and would usually contain common settings determined by the system administrator such as search paths, formats, libraries, and so on. This file cannot

      normally be altered by individual users.
♦   user home directory - unique for each user (Unix).
♦   startup directory - the working directory when Creo starts up[1].

Settings made in the first copy (*config.sup*) are locked and cannot be overridden by users. This is handy for making configuration settings to be applied universally across all users at a Creo installation site (search paths for libraries or drawing templates, for instance).

During the installation procedure for the software, a copy of a stripped-down *config.pro* file was created in the **../text** directory of the software loadpoint. The contents of this file depend on which system of units was chosen (English or metric) during software installation. If metric was chosen, a number of options are set to reflect ISO standards. If English was chosen, these options are set to ASME and ANSI standards.

An individual user can modify entries in the last two copies of *config.pro* to suit their own requirements. If the same entry appears more than once, the last entry encountered in the start-up sequence is the one the system will use. After start-up, additional configuration settings can be read in at any time. These might be used to create a configuration unique to a special project, or perhaps a special type of modeling (sheetmetal, for example).

The Configuration File Editor

You can access your current configuration file using

File ➤ Options ➤ Configuration Editor

This brings up the **Options** window with the existing *config.pro* contents shown at the right. See Figure 4. The **Show** field indicates the source of the currently listed options.

Click the **Show** pull-down at the right, and select **All Options**. A complete list of all the Creo configuration options will appear. The first column shows its name, and the second column shows its current value. A value with an asterisk indicates a default value. The third column indicates its status (a solid green circle means the option has been read from a *config* file, rather than the system default), followed by a one-line description in the fourth column. Note that you can resize the column widths by dragging on the vertical column separator bars at the top of the display area.

Browse down through the list. There are a lot of options here (>1000!). Note that the options are arranged alphabetically. This is because of the setting in the **Sort** pull-down menu in the top-left corner. Change this to **By Category**. This rearranges the list of options to group them by function. For example, check out the settings available in the **Environment** and **Sketcher** groups. The list of options is a bit overwhelming.

---

[1] In Windows, right click on the Creo icon on the desktop (if it exists), select **Properties ➤ ShortCut** and examine the **Start In** text entry field.

Fortunately, there are a couple of tools to help you find the setting you're looking for. Let's see how they work.

Open the **Show** list again and select **Current Session** and select *Sort(Alphabetical)*. There will likely be some options listed, if not from a *config.pro* in your startup directory, then from the software loadpoint. You can see these by selecting from the options in the **Show** list. In that list, selecting *Only Changed* makes it easier to see new settings.
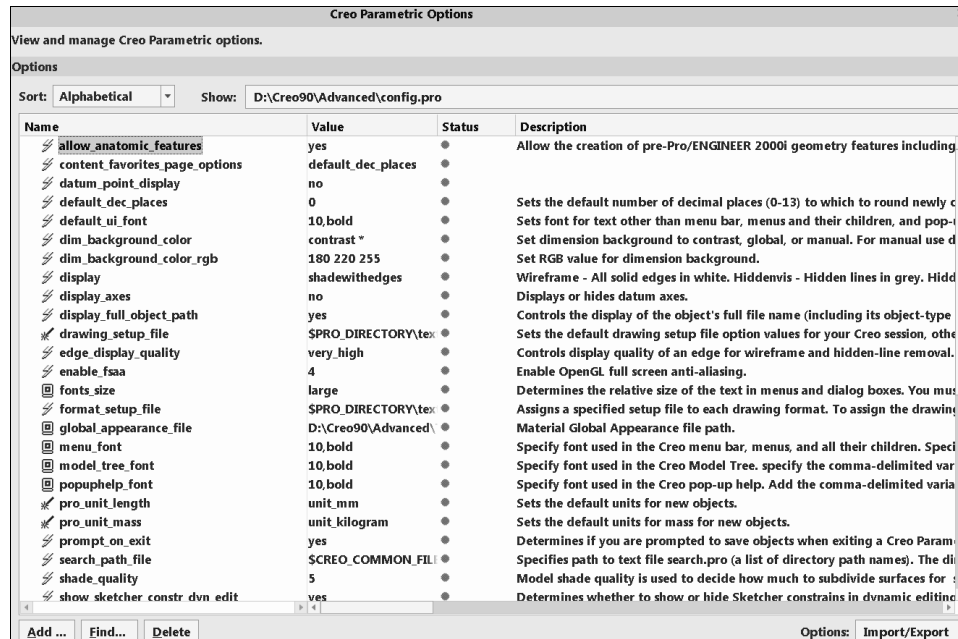

Figure 4 The *config.pro* file layout showing current settings

Adding Settings to *config.pro*

Let's create a couple of useful settings. At the bottom of the **Options** window are two buttons: *Add* and *Find*. If you know the name of the option, you can just select *Add* and type it in the top box shown in Figure 5.
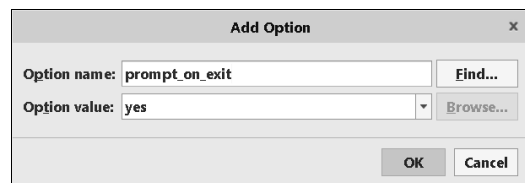

Figure 5 The *Add* window for config options

For new users, a useful setting is the following. In the name box enter the option name *prompt_on_exit*. As you type this in, notice that Creo anticipates the rest of the text based on the letters you have typed in. After typing enough characters (up to the "x" in "exit" - it takes Creo that long to find out you don't want *prompt_on_erase*), the rest of the desired option will appear. In the pull-down list beside **Option Value**, select *Yes* (or just type "y"). Note that the option name is not case sensitive and the default value is indicated by an asterisk in the pull-down list. Now select the *OK* button. The entry now appears in the data area. A bright green star in the **Status** column indicates that the option has been defined but has not yet taken effect.

Now enter a display option. The default part display mode in the graphics window is
Shaded. Some people prefer to work in shaded mode with part edges shown - let's make
it the default on start-up. Once again, we will enter the configuration option name using
Add and pick the value from a drop-down list. The option name and value we want are

> **display**   **shadewithedges**

Now select *OK* as before. Add the following option to control how datum planes are
displayed during dynamic spinning with the mouse:

> **spin_with_part_entities**   **yes**

Another common setting is the location of the Creo trail file. As you recall, the trail file
contains a record of every command and mouse click during a Creo session. The default
location for this is the start-up directory. Theoretically, trail files can be used to recover
from disastrous crashes, but this is a tricky operation. Most people just delete them. It is
handy, therefore, to collect trail files in a single directory, where they can be easily
removed later. There is an option for setting the location of this directory. Suppose we
don't know the configuration option's specific name. Here is where a search function
will come in handy.

At the bottom of the **Options** window, click the
Find button (you may have noted that this is also
available in the **Add** window, Figure 5). This
brings up the **Find Option** window (Figure 6).

Type in the keyword *trail* and leave the default
Look in(ALL_CATEGORY) (note the check box
option to search the descriptions for the keyword;
we don't need that in this case) then select *Find
Now*.

Several possibilities come up. The option we
want is listed as **trail_dir** - scroll the description
to the right to confirm this. Select this option and
then pick the *Browse* button at the bottom to
identify a suitable location on your system for the
value, perhaps something like *d:\temp*. Then
select *Add/Change*. The new entry appears in
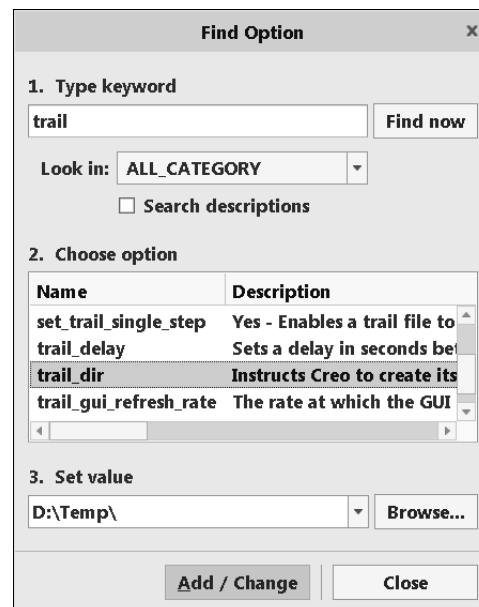the **Options** window. In the **Find Option**
window, select *Close*.

Figure 6 Using *Find* to locate an option

For some options, the value is numeric, for example setting a default tolerance, number
of digits, or the color of entities on the screen. Colors are typically given using three
integers representing RGB composition. In these cases, you can enter the relevant
number (or numbers separated by either spaces or commas).

For practice, enter the following options, either directly or using *Find*. (HINT: use the RMB pop-up menu anywhere in the option table and select *Add*.) The order that the configuration options are declared does not matter. Feel free to add new settings to your file (for search paths, libraries, default editors, default decimal places, import/export settings, Sketcher preferences, and so on).

allow_anatomic features	**yes**
default_dec_places	**1**
display_full_object_path	**yes**
sketcher_starts_in_2D	**yes**
sketcher_lock_modified_dims	**yes**
dim_background_color	**manual**
dim_background_color_rgb	**250 250 150**

In the Show pull-down list, select *Current Session*. Notice the icons in the first column beside the option names. These mean the following:

  ⚡ (lightning) - option takes effect immediately
  ✦ (wand) - option will take effect for the next object created
  ▣ (screen) - option will take effect the next time Creo is started
  ⊘ (illegal) - option is illegal or not recognized

If you are using a *config* file from a previous version of Creo you may see the red illegal symbol, which means that the option is no longer used.

Try to add an illegal option name. For example, in a previous release there was an option **sketcher_readme_alert**. Try to add that and set it to **Yes**. When you try to set a value for this, it will not be accepted (the *OK* button stays gray). Creo only recognizes valid option names! Thus, if you mistype or enter an invalid name, this is indicated by not being able to enter a value for it.

Deleting Configuration Options

Highlight one of the options (maybe an illegal one) and use the RMB pop-up to select *Delete*. Save the modified file with *Export*, then use *OK* to close the window.

Saving Your *config.pro* Settings

The green stars in the **Status** column indicate the new settings have not taken effect yet. To store the settings we have just created, select either the *Export Configurations* button, or open the *Import/Export* menu of the **Options** window. In the **Save As** window that opens (make sure it is showing the start-up directory), accept the default file name *config.pro* and select *OK*.

Loading a Configuration File

You may have noticed (in the Import/Export menu) the *Import Configuration* command.

Select that now and pick the desired file and then ***Open***. These settings will be read in and added to the current ones but not activated immediately (note the green star). That will happen when you close the window with ***OK***.

Checking Your Configuration Options

Because some settings (mainly dealing with windows properties like fonts and colors) will not activate until Creo is restarted, many users will exit after making changes to their *config.pro* file and then restart, just to make sure the settings are doing what they are supposed to. Do that now. This is not quite so critical since the **Options** window shows you with the lightning/wand/screen icons whether an option is active. However, be aware of where Creo will look for the *config.pro* file on start-up, as discussed above. If you have saved *config.pro* in another directory than the one you normally start in, then move it before startup. On the other hand, if you have settings that you only want active when you are in a certain directory, keep a copy of *config.pro* there and load it once Creo has started and you have changed to the desired directory. To keep things simple, and until you have plenty of experience with changing the configuration settings, it is usually better to have only one copy of *config.pro* in your startup directory.

Note that it is probably easier to make some changes to the environment for a single session using ***File ➤ Options***. Also, as is often the case when learning to use new computer tools, don't try anything too adventurous with *config.pro* in the middle of a critical part or assembly creation session - you never know when an unanticipated effect might clobber your work!

Customizing the Toolbars

For the following, you will have to load one of the parts you have previously made. Do that now.

Click the right mouse button on the Graphics toolbar. This brings up a menu like the one shown in Figure 7. This allows you to select which commands appear on the toolbar, where the toolbar is located, the size of the toolbar icons, or reset to the default toolbar.

The option ***Propagate Customization*** will keep the same toolbar across all modes of Creo (into Simulate, for example).

Now use the RMB pop-up on the Quick Access toolbar at the top. A pop-up will allow you to specify the location (relative to the ribbon) or remove it entirely, or modify the commands. If you select
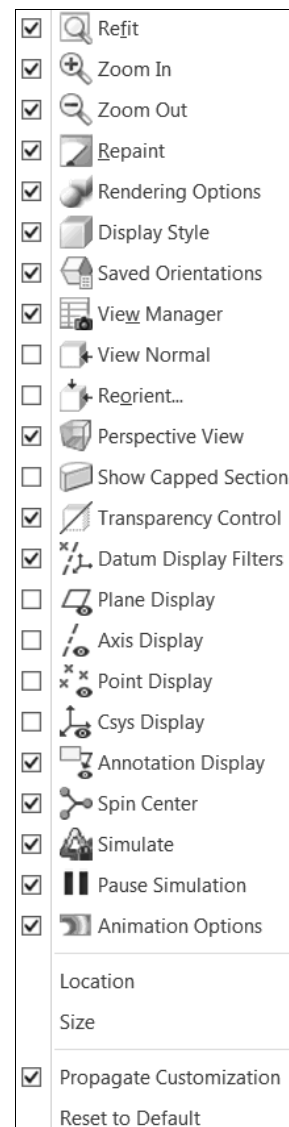

Figure 7 Customizing the **Graphics** toolbar

> *Customize Quick Access Toolbar*

you will be taken to the regular Creo Parametric Options window. Here you may select from any command in the left pane and *Add* it to the toolbar list on the right. You can change its position relative to the other commands using the up/down arrows at the right side of the window. Notice that there is a button that will reset to all default values. Also, be aware that some commands will not be available unless you are in relevant modes (for example, the Model Colors command is not available unless a model is loaded). Close this window. Any changes you made were saved automatically and will be loaded again at startup.

Customizing Ribbon Tabs and Groups

With the mouse on the ribbon, use the RMB and select *Customize the Ribbon*. This opens the usual **Options** window arranged as shown in Figure 8. The Creo commands are listed in the center pane, and the ribbon structure is shown in the right pane. Each of the tab titles and groups can be expanded to show the commands it contains.
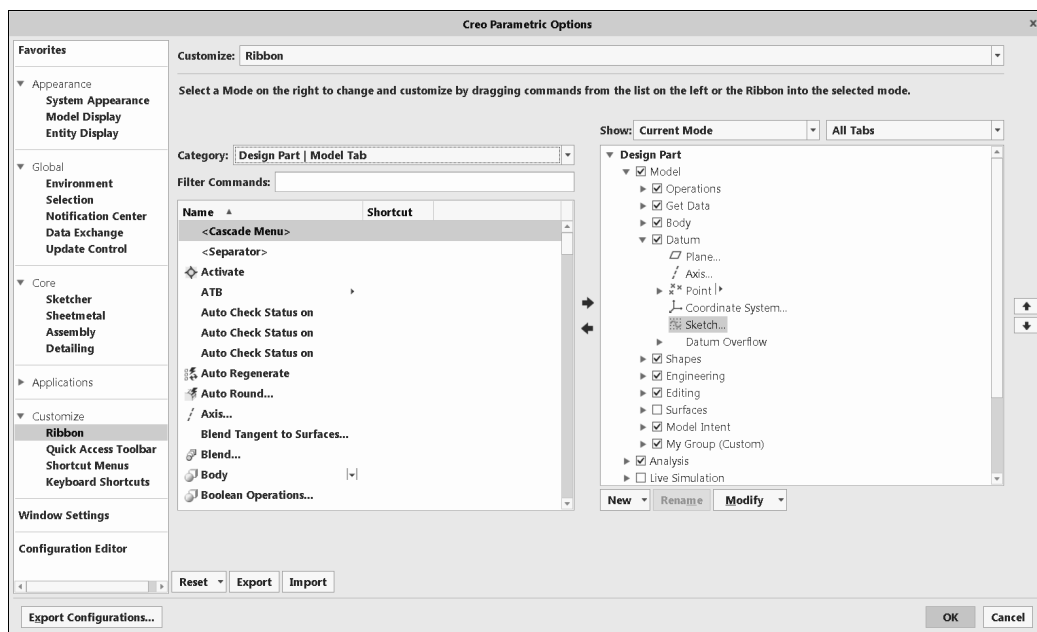

Figure 8 Customizing the Creo Parametric ribbons

We can do a lot to modify the contents and appearance of the ribbons. First, each tab and group has a checkbox to specify whether it should be displayed. Within each group, we can modify the appearance of the command icons. For example, select the *Mirror* command in the *Editing* group of the *Model* tab. Now you can select the *Modify* command at the bottom to change the size and appearance of the command. It is also possible to do this a bit more easily by moving the **Option** window out of the way and just selecting the button in the ribbon and using the RMB pop-up menu. Depending on how much space there is on your screen, you may find that the appearance of other commands in other groups may change (mainly the command label).

Select the **Model** tab, then using the RMB menu select *Add New Group*. We can specify a new name with *Rename*. Now with the new group selected, we can move commands from the left pane into the new group. This allows you to collect commands that you use often that are located on different ribbons. Create the group shown in Figure 9. You can remove commands, groups, and tabs using the RMB pop-up menu and selecting *Remove*.

Figure 9 A customized ribbon group

These settings for the user interface will be saved automatically in the file *creo_parametric_customization.ui* at the Creo program loadpoint. If you have specific settings you want for a specific project, save the file in the start-up directory for that project using **Export**. To reload them later, use **Import**.

Helpful Hint

It is tempting, if you are blessed with a lot of screen space, to over-populate the toolbars and ribbons by trying to arrange every commonly used command on the screen at once. Before you do that, you should work with Creo for a while. You will find that Creo will generally bring up the appropriate toolbars for your current program status automatically. For example, if you are in Sketcher, the Sketcher short-cut buttons will appear. Thus, adding these buttons permanently to any toolbar is unnecessary and the buttons will be grayed out when you are not in Sketcher anyway - you are introducing screen clutter with no benefit. Furthermore, many commands are readily available in the right-mouse pop-up menus. Since these are context sensitive, you will only have to choose from commands that are useful at that moment.

There is lots more information in the on-line help. A good place to start is the following pages in the Help Center:

> **Fundamentals**
> > **Fundamentals**
> > > **Fundamentals Overview**
> > > **Configuring Fundamentals**
> > > **User Interface Basics**
> > > **Creo Parametric User Interface**

You can also do a search in the Help pages for *config.pro*.

Keyboard Shortcuts - Hotkeys

The table at the right shows some of the default hotkeys. These are shortcuts to commonly used commands. As indicated, some of these are launched by holding down the *Ctrl* key while the keyboard letter is pressed. Most of these are mnemonic (indicated by the *__highlighted__* letter). Some others involve a single keyboard key (examples: *P* creates a datum plane, *X* creates an extrude, *R* creates a round). To see the other keyboard shortcuts, go to the **Options** window and select

Keyboard Shortcuts

in the **Customize** area. In the **Show** pull-down list, select **Commands with assigned shortcuts**. You can also look for these single letter commands in the button pop-ups that show when you hover the mouse over a command in the ribbon.

Key	Command
Ctrl+A	*A*ctivate window
Ctrl+C	*C*opy feature
Ctrl+D	*D*efault view orientation
Ctrl+G	re*G*enerate model
Ctrl+N	create *N*ew model
Ctrl+O	*O*pen existing model
Ctrl+P	*P*rint
Ctrl+R	*R*epaint graphics window
Ctrl+S	*S*ave file
Ctrl+V	paste feature
Ctrl+Y	redo
Ctrl+Z	undo

Keyboard Shortcuts - Mapkeys

A mapkey is a short sequence of keyboard key strokes or a function key that will launch one or, more commonly, an extended series of Creo commands. Since many simple commands are launched using toolbar buttons, mapkeys are typically used to start extended command sequences. Mapkeys are very similar in usage to macros that can be defined in other software packages. Mapkey definitions are contained/included in your *config.pro* file, so they are loaded at start-up.

The mapkey can be as long as you want; most users restrict mapkeys to only 2, or sometimes 3, lower case characters. This gives several hundred possible mapkey sequences - more than you can probably remember effectively. Creo constantly monitors the keyboard for input and will immediately execute a defined command sequence when its mapkey is detected. Single character mapkeys should be avoided due to the way that Creo processes keyboard input. If you have two mapkeys "v" and "vd", for example, the second mapkey would never execute since Creo will trap and execute the first one as soon as the "v" is pressed.

Ideally, you would like to have mapkeys that are very easy to remember, like "vd" (view default), or "tc" (toggle colors). Furthermore, mapkeys are usually launched with your opposite hand from the mouse. Because it is common to only use two characters, it will

take some planning to decide how you want to set up your definitions to use only a couple of easy-to-remember key strokes! The mapkey should be mnemonic but can't collide with other definitions. You don't want to have to remember that "qy" means "repaint the screen."

A practical limit on usable mapkeys is perhaps in the range of 10 to 20, although some "power users" can use many more. In previous versions of the software, some commands used to be several menus deep into the program, requiring numerous mouse clicks, so mapkeys made life a lot easier. However, with the development of the Creo interface, the need for dozens of mapkeys is diminishing - you can probably get by with just a few (or even none!).

Listing Current Mapkeys

To see a list of your current mapkeys (some may be defined in your *config.pro* file) select

> *File > Options*
> *Environment*
> *Mapkeys Settings*

This dialog window (Figure 10) allows you to define and record, modify, delete, run, and save mapkeys. Note that each mapkey has a short **Name** and **Description**. The **Name** will be used on any short-cut button (described below), and the **Description** will appear in the message area in the main graphics window. Mapkeys that start with a "$" are function keys.

Mapkeys created using a previous release of Creo may differ in command syntax and it is likely that some mapkey definitions from previous releases will not function properly. However, mapkeys are easy enough to record.

Figure 10 Defining mapkeys

In the following, it is assumed that you have no mapkeys defined as yet. If any of these tutorial mapkeys collide with existing mapkeys shown in the mapkeys list (Figure 10) or the built-in keyboard shortcuts, you can modify the keyboard sequence (for example, use "vv" instead of "vd") for the new mapkey.

Creating Mapkeys

New mapkeys are created as follows. We will create a very simple mapkey sequence "vd" that will reorient the view to the default orientation and, from wherever you are in the ribbons, return you to the **Model** ribbon. To set this up, you will have to bring in one
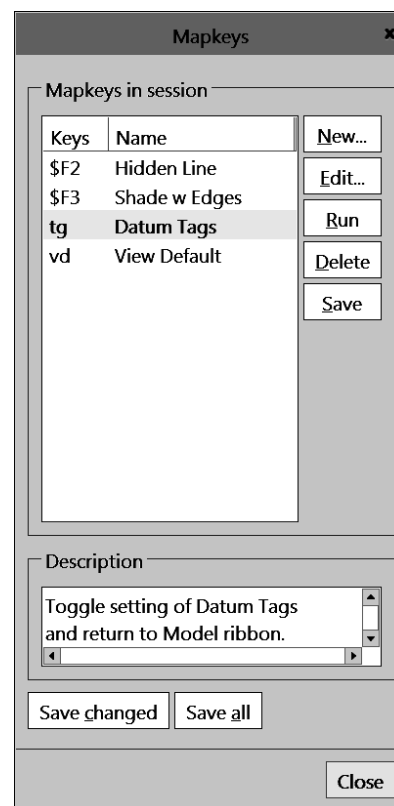
of your previously created parts. Do that now. We will not be modifying the part.

Select the *New* button in the **Mapkeys** menu (Figure 10). The **Record Mapkey** dialog box shown in Figure 11 will open. Enter the data shown in the figure: key sequence, name, description. Now we record the command sequence while we execute the desired command sequence:

> *Record*

Click on the **View** ribbon tab and select *Standard Orientation*. Now click on the **Model** ribbon tab. Finally, in the **Record Mapkey** dialog window select

> *Stop > OK*

Every command and keystroke between *Record* and *Stop* is added to the mapkey. It's that easy! Spin the model with the middle mouse button[2] and select any other ribbon. In the mapkeys window, highlight the new mapkey "vd" and select the *Run* button. You should end up in default view orientation back in the **Model** ribbon. It's a good idea to check your mapkey definitions now when it is easy to modify them.

As mentioned above, mapkey definitions are saved in a configuration file (as in *config.pro*). New mapkey definitions are appended to the end of the file. If you redefine a mapkey (or use a duplicate keystroke sequence), the definition closest to the bottom of the *config* file is the one that will be used. There are three ways to save the mapkeys using the buttons in the **Mapkeys** window:

Figure 11 Creating a mapkey

> *Save* - saves only the highlighted mapkey
> *Save Changed* - saves any mapkeys changed this session
> *Save All* - saves all mapkeys defined for the current session

Select one of the three options and save our "vd" mapkey.

Also, be aware that if you save all mapkeys defined in the session, they are appended to the end of the *config.pro* file. If you do this excessively, the file can become quite large (and slow to load). You should occasionally edit the file with a text editor to remove the early duplicates.
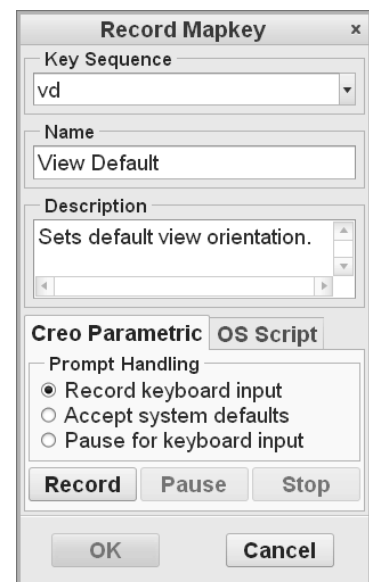
---

[2] Unfortunately, this may cause the Mapkeys window to move behind the graphics window. Just open it again as above.

Close the **Mapkeys** window. Minimize Creo and open *config.pro* using your system text editor. Scroll down to the bottom of the file to see the new lines that describe the mapkey - it will extend over several lines. It is possible to move the mapkey definitions elsewhere in the file, but for each definition these lines should never be separated since they are a continuation of the same sequence. It is possible, but probably not advisable, to try to edit the mapkey definitions manually - leave that to the power users! Exit your text editor and restore the Creo window.

Some final points about mapkeys: it is possible to set up the mapkey so that execution will pause to allow user input during the command sequence, either by picking on the screen or through the keyboard. Mapkeys can also call other mapkeys. You might like to experiment with these ideas on your own. The possibilities for customization are almost limitless! As mentioned earlier, with the "flattening" of the user interface in Creo, the need for mapkeys is greatly diminished. They are primarily used as shortcuts for frequently used sequences of multiple commands.

Creating a Customized Part Template

Most part files that you create contain many common elements such as datums, defined views, coordinate systems, parameters, and so on. Creating these from scratch for every new part that you start is tedious and inefficient. In Creo's predecessor a very handy model creation tool used the notion of a "start part" which contained these common elements. Users would then create a mapkey that would bring the part into session and then rename it. This made the creation of new parts very quick and efficient, with the added bonus that standard part setups could be employed.

This "start part" functionality has been built into the program using part templates. Several part templates are included with a standard Creo installation for solid and sheet metal parts in different systems of units. You may have a reason at some point to create your own template, which we will do here. We'll also set up the system so that your template becomes the default when you create a new part. Then you can immediately get on with the job of creating features. We will create the custom part template using one of the existing templates as a basis. This demonstrates that any part file can be used as a template, even ones containing existing solid features.

Select *File > New* (or Ctrl-N). Make sure the **Part** and **Solid** radio buttons are selected. Deselect the **Use Default Template** box, and enter a name [*mytemplate*]. Select *OK* and in the next window, select the **mmns_part_solid_abs** template. Notice that it has two parameters (MODELED_BY and DESCRIPTION). Fill in the MODELED_BY parameter with your name or initials. Select *OK*.

Now we will customize the existing part. You may find that you don't often use the default part coordinate system, so delete that. If you do a lot of axisymmetric parts, you might like a default axis. Create one now by selecting (with Ctrl) the RIGHT and FRONT datums, then the *Axis* command. The part should look like Figure 12.

Let's assign a material to the part. Select *File >
Prepare > Model Properties*, then beside **Materials**
select *change*. Go to the legacy material library and
double click on **steel.mtl**. Select *OK > Close*.

Open the **Materials** folder under **Design Items** in the
model tree to see the material definition. Select it and
in the RMB pop-up select *Edit Definition*. Note the
default values and units for material properties.

The normal views (FRONT, RIGHT, etc) are already
defined. Try creating your own custom views (using
Reorient) like left and right isometric views.

Figure 12 New template file
mytemplate.prt default datums

We are finished with creating the part, so save it in your default working directory with
the name **mytemplate.prt**. If you have write access, copy the part file to the Creo
installation TEMPLATES directory, something like

         <Creo Loadpoint>\common files\templates

This is the default directory where Creo will look for part templates. If you do not have
write access to this directory, leave the part file in your working directory. You can
rename the file to remove the version number if you want, so that it appears as
mytemplate.prt rather than **mytemplate.prt.1**.

Setting the Default Part Template

We can tell Creo to use our new template as the default by setting an option in
config.pro. Select

         File > Options > Configuration Editor

and *Add* (or edit) the option **template_solidpart**. Set the value for the option by
browsing to the template directory (or use the current working directory, wherever you
have saved the template file) and selecting the part file **mytemplate.prt** we created
above. Alternatively, you can specify the location in the system templates directory using
(type this exactly as shown)

         template_solidpart $PRO_DIRECTORY\templates\mytemplate.prt

Before you exit the options window, don't forget to store your new *config.pro*.

Using the New Part Template

Close the current part window and select *Erase_Not_Displayed*. Select
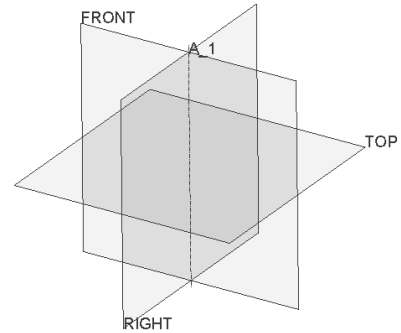
         File > New

Make sure the **Use Default Template** box is now checked, enter a name (like *test*), and select *OK*. A copy of your custom template (check for the axis, the assigned material, and the named views) is now brought into session and given the name you specified.

Adding Mapkeys to Ribbons

Any Creo command or mapkey can be added to any of the existing ribbons and toolbars. Let's add a mapkey to the group **My Group** that we created in the **Model** ribbon earlier in this lesson. Select

>   *File ➤ Options ➤ Ribbon*

In the dropdown **Category** list, select **Mapkeys**. Pick the *View Default* mapkey that we made earlier, and *Add* that to **My Group** in the **Model** ribbon group list at the right. Below the list select *Modify* to edit or change color of the button image, or use *Choose From Existing* icons to replace default button image if you want. See Figure 13.

Now, of course, you will only see the **My Group** buttons when the **Model** ribbon is open. To make the mapkey available at any time, it needs to go in the Quick Access toolbar at the top. Select

>   *File > Options*
>   *Quick Access Toolbar*

Once again select **Mapkeys** in the pulldown list, then pick the **View Default** mapkey and move it to the list at the right with the right arrow. You can position the icon on the toolbar using the up and down arrows at the far right. Close the **Options** window. Your settings will be saved for the next time you launch Creo.
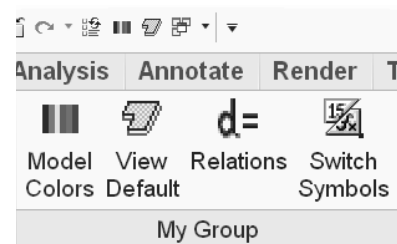
Figure 13 Mapkey added to ribbon group and Quick Access toolbar

We have spent a lot of time so far looking at how to set up Creo to help you be more efficient and productive with your time. It's time to actually produce something! We'll take this opportunity to explore the concept of multibody modeling (introduced in Creo Parametric 7.0). This provides yet another powerful and flexible tool to assist in model creation. Now is a good time to take a break.

Introduction to Multibody Modeling

In normal part creation, new features are created and added automatically at the bottom of the model tree (unless the insertion point has been moved). Features that produce solid geometry (extrusions, revolves, rounds, and so on) will modify the existing solid by

either adding or subtracting from the solid volume of the part[3]. Solid features that overlap with the existing geometry are automatically merged with it. If the same geometry is declared to be a cut, then some of the existing part is removed. The point being that, in traditional modeling, there is a single solid body within the part model and the part is the same designated material throughout.

Multibody modeling allows for the presence of several bodies within a single part. When creating solid geometry, Creo allows new features to be assigned to different bodies within the part. The bodies can then be used in Boolean operations (***Merge***, ***Intersect***, ***Subtract***) that come out of a branch of modeling called Constructive Solid Geometry (CSG). Multibody modeling is basically a way of organizing features in closely related or functional groups. Note that only solid geometry is included in a body - no datum features, sketches, or curves. As features are created, they can use previously created features and geometry in any body in the part for references. Bodies within a single part can be assigned different materials. Bodies can become either permanent parts of the model or be used as tools and to modify other bodies. Multibody modeling is similar to creating new parts within an assembly but without having to worry about complicated assembly constraints or references. Using multibody modeling is a way to organize feature creation that offers additional flexibility in model creation.

We will have a very quick look at how bodies are created and used with basic Boolean operations, how to create new parts from bodies, how bodies appear in the model tree.

The model we are going to create is shown in Figure 14 - we'll call it the *podracer*. It consists of two parts: the *cockpit* and the *canopy*. This simple assembly derives from a single master part file containing (when we are done!) two bodies that are used to create two separate (but associated) part files.
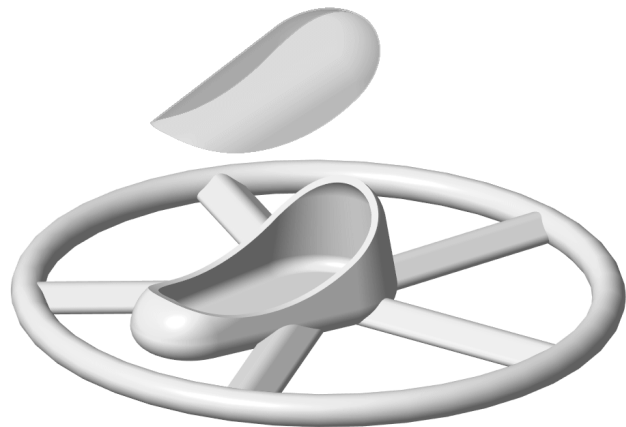

Figure 14 Final assembly (exploded) created using multibody modeling

Using the part template we just created, start a new part called ***mbody***. It should have a vertical datum axis at the intersection of FRONT and RIGHT. In the model tree **Design Items** folder, notice that the material is defined and there is currently one body.

The first feature is an extrude coming off the TOP datum. Use the sketch shown in Figure 15 (sketch references are FRONT and RIGHT). Set the depth to **200**. In the Protrusion dashboard, open the **Options** tab and check the box beside ***Add Taper***. Set the taper to **10**° as shown in Figure 16. Accept the feature. It is helpful when dealing with multibody models to assign different colours to each body to distinguish them. Do this as soon as

---

[3] Unless, of course, they are buried features, in which case they will do nothing.

each body is created, like now. Make sure you assign the colour just to the body and not the entire part (use the Body listing in the Bodies folder). Place an **R25** round on the lower edge.
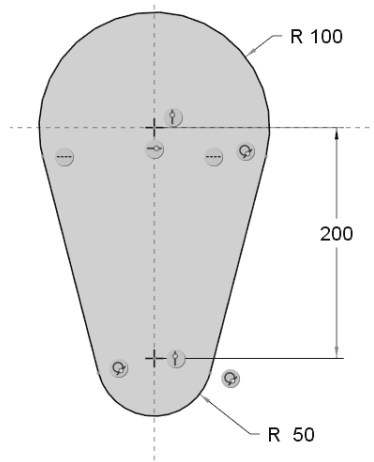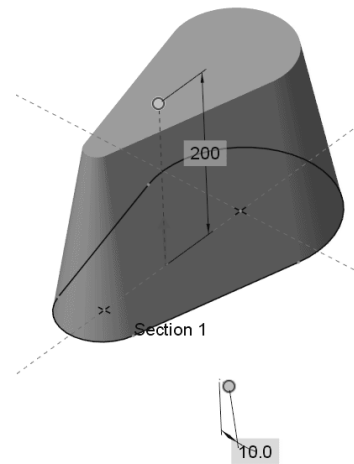

Figure 15 Sketch for first body

Figure 16 Preview of first body with taper

Launch the ***Extrude*** command again, select FRONT as the sketching plane, and make the sketch shown in Figure 17. Set a blind depth of **350** and reverse the direction of the extrude so that it is coming off the negative side of FRONT. In the Protrusion dashboard, open the **Body Options** tab and check the box **Create new body**. (You may have to toggle the ***Remove Material*** button.) See Figure 18.
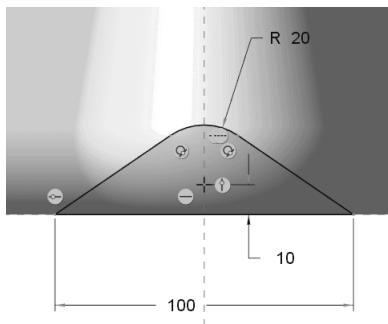
Figure 17 Sketch for second body
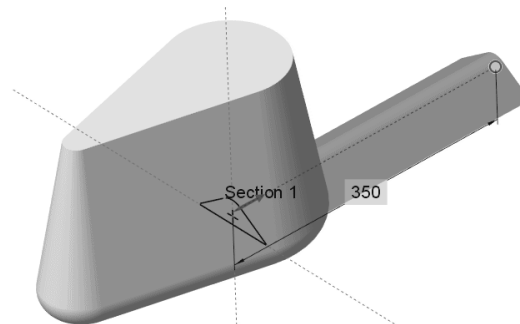
Figure 18 Preview of second body

Check the model tree to see that Body 2 has now been added in the **Bodies** folder and is now the default body. The extrusion is listed at the bottom of the model tree. Assign an appearance to the new body then create an **R8** round on the two edges of Body 2.

Let's explore the model tree. Open that and expand the **Bodies** folder near the top. Take a moment to add (if they are not there already) the columns shown in Figure 19.

	Feat #	Feat Type	Feat Subtype	Contributes To	Feature Status
MBODY.PRT					
▼ Design Items					
▼ Materials	<None>				
└ →STEEL	<None>				
▼ Bodies (2)					
▼ Body 1					Contains Geometry
└ STEEL	<None>				
Extrude 1	5	Protrusion	Extrude	Body 1	
Round 1	6	Round	General	Body 1	
▼ Body 2					Contains Geometry
└ STEEL	<None>				
Extrude 2	7	Protrusion	Extrude	Body 2	
Round 2	8	Round	General	Body 2	
RIGHT	1	Datum Plane			
TOP	2	Datum Plane			
FRONT	3	Datum Plane			
A_1	4	Datum Axis			
▶ Extrude 1	5	Protrusion	Extrude	Body 1	
Round 1	6	Round	General	Body 1	
▶ Extrude 2	7	Protrusion	Extrude	Body 2	
Round 2	8	Round	General	Body 2	

Figure 19 Model tree for part with multiple bodies and added columns

Notice that the bodies have been assigned the part material automatically. That can be overridden (Hint: RMB on the Body). Other than the material, the bodies contain only solid features, which are actually listed twice in the model tree. The bodies will have an icon that depends on their status. So far, both our bodies contain solid features. You can expect to see the following (select *Tree Filters ➤ Body/Quilt*):

⬜ Body containing geometry
⬜ Body containing no geometry
⬚ Body containing no features
⬜ Consumed Body
⬚ Construction Body
⬆ (Any icon with a star) Default Body

This is also indicated in the **Feature Status** column of the model tree. Check the model tree periodically as we proceed through the creation of this model to see the changes in status. Feel free to toggle the display of columns on and off using *Show/Hide Columns*.

Click Body 2 in the model tree (or in the graphics window. Hint: set the selection filter to Bodies). A mini-toolbar appears as in Figure 20. Take a moment to identify the various commands on this toolbar - it will save you time later searching for them in the dashboard. If you click on a non-default body (without the star) in the model tree, the mini-toolbar will also have a command (a star - no surprise!) that will make the selected body the default body.
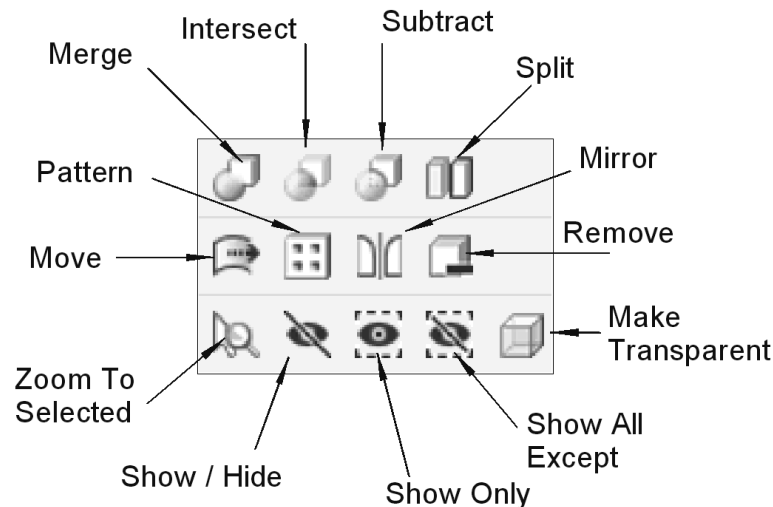
Figure 20 Mini-toolbar for multibody operations

With Body 2 selected and highlighted in green
in the graphics window, in the mini-toolbar
select *Geometry Pattern*. In the **Pattern**
dashboard, note the available pattern types and
change the pattern type to **Axis**, pick on the
datum axis A_1, set the **Number of members**
to **5**, and click the **Angular Extent** button for a
360° pattern. Open the pattern folder in the
Design Items area. Note that the material of the
pattern leader has been automatically copied to
the new bodies (3 through 6) but not the
appearance (see the graphics window). Also
check the pattern listed at the bottom of the
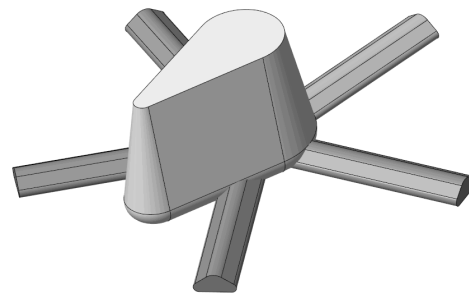model tree (what are its contents?).

Figure 21 A pattern of bodies

What happens if you try to change the radius of the (**R8**) round by picking on any of the
copies? How can you change the radius value? This behaves a bit differently from normal
patterned features, where there is a bidirectional associativity between instances.

Boolean Operations

We will now perform our first Boolean operation to merge the five arms into a single
body. In the **Body** group in the model dashboard, select

> *Boolean Operations*

This opens the **Boolean Operations** ribbon. It contains the three CSG operations *Merge*
(the default), *Intersect*, and *Subtract*. The default is exactly what we want. The **Body to
modify** is what the other bodies will be merged to, in our case pick on Body 2 (the first
arm). Click the collector for **Modifying bodies** to activate it and then CTRL-click on
each of the other four arms. The **References** panel in the dashboard shows all the bodies.

Accept the merge operation with *OK*.

Check what has happened to all the bodies in the model tree. What icon is in front of bodies 3 to 6? What is their **Feature Status**? Come back later to see if they can be *Removed* with impunity. In the Bodies folder, what features are listed under Body 2?

Turn the model over to see that Body 1 and Body 2 remain distinct (easy to see if they are different colours), even though they overlap. This would not happen with regular features. In passing, you might note that you can rename a body (since numbers will eventually get confusing!) using the *Rename* command in the RMB pop-up menu with the body selected. What other commands are available in the RMB menu?

We are now going to cut away the portion of the arms (Body 2) that overlaps with Body 1. This is so that we can eventually shell out Body 1 and not have the arms protrude inside the body. Select

Boolean Operations ➤ Subtract

The **Body to modify** is body 2 (the merged arms) and the **Modifying body** is body 1. There is an option **Keep bodies** for the modifying body. In this case, we want to keep body 1, so select the option. Accept the operation with *OK*.

Select Body 1 in the model tree and in the mini-toolbar, select *Transparent*. What you will see is the merged arms with all geometry overlapping with Body 1 removed. See Figure 22. It is interesting to note that body 2 contains several disjointed solid pieces. The notion of a body is evidently logical rather than physical. In the Graphics Toolbar, open the *Transparency Control*, and turn off transparency of **Solid Bodies**.
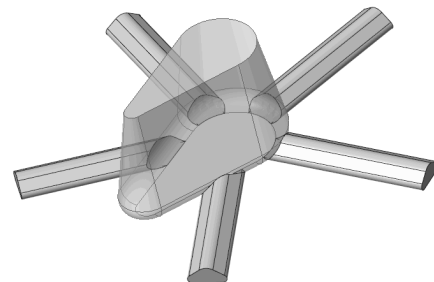
Figure 22 Merged Body 2 after Subtracting Body 1

To see a variation of the *Subtract* operation, we'll now create a new body that will be used to remove a portion on top of Body 1. Create an extrusion (sketch on RIGHT as shown in Figure 23). Make it *Symmetric* with a depth of **400**. In the **Body Options** panel, check the box to **Create new body**[4]. Accept the feature (Figure 24) with *OK*. It will form Body 7.

---

[4] If you forget to do this, you can always use *Edit Definition* in the model tree, which brings you back to the dashboard.
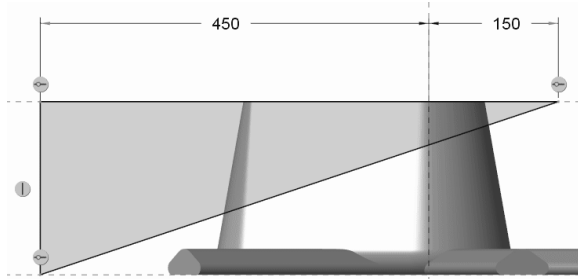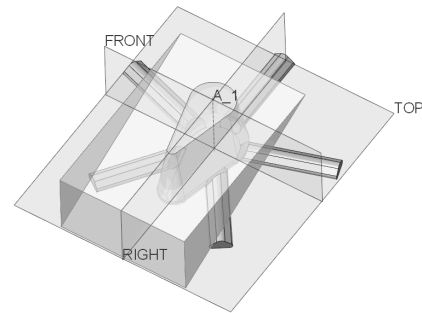
Figure 23 Sketch for Body 7

Figure 24 Body 7 created

Launch the next Boolean operation:

Boolean Operation ➤ *Subtract*

The **Body to modify** is Body 1. The **Modifying body** is Body 7 that we just created. In this case we did not want to keep the modifying body so leave that option alone. Accept with *OK* and observe what has happened in the model tree. What is the **Feature Status** of the Body 7?

Check that body 1 is the default and create an **R50** round on the top edge. See Figure 25.
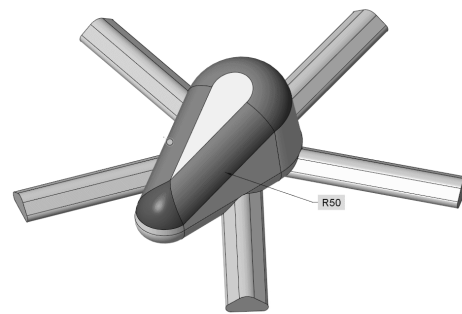
Figure 25 Round added to Body 1

We are now going to do a couple of related Boolean operations that will create a canopy and an exactly matching opening on what will be the body of the racer.

Start a new extrude and create the sketch shown in Figure 26. This will be a *Symmetric* protrusion with a depth of **300**. Don't forget to open the **Body Options** tab and check **Create new body**. This will be Body 8.
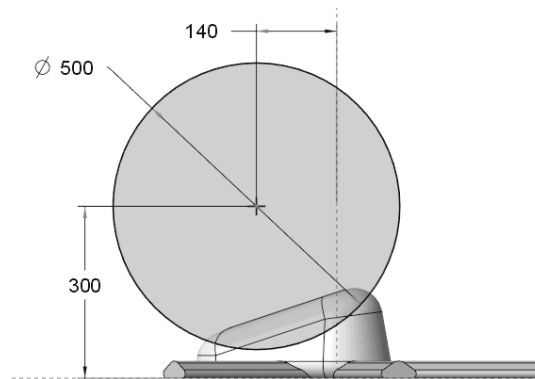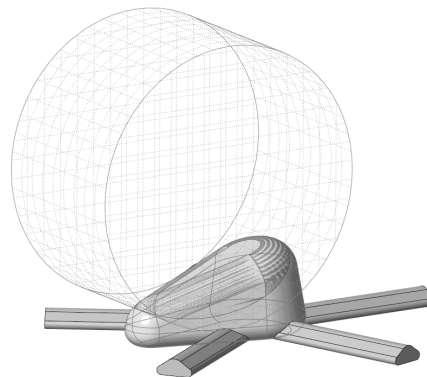
Figure 26 Sketch for Body 8

Figure 27 Using *Intersect* to create the canopy

Now we'll create the canopy.

Boolean Operations ➤ *Intersect*

The **Body to modify** is Body 8; the **Modifying body** is Body 1. Select the option to *Keep bodies*, since after the intersection is found we still want Body 1 in the model. Accept the operation with *OK*. See Figure 27. Observe the model tree.

Now subtract the canopy body from Body 1:

Boolean Operations ➤ *Subtract*

The **Body to modify** is Body 1; the **Modifying body** is Body 8. Select **Keep bodies** and accept the feature with *OK* . Assign an appearance (make it transparent) to the canopy as in Figure 28.
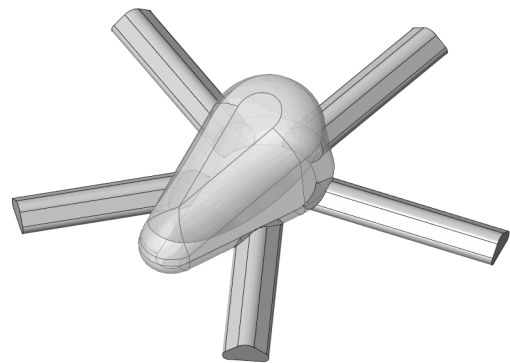
Let's explore the model tree a bit. Select Body 1 and in the mini-toolbar make it the default body (with the star), then *Hide* it, again using the mini-toolbar. This shows that the default body does not actually have to be in view.

Figure 28 *Canopy* body subtracted from *cockpit* body

Pick the canopy in the model tree and in the mini-toolbar select *Show Only* and make it the default body. Now launch the *Shell* command. In the **References** panel make sure only Body 8 is selected, activate the **Remove surfaces** collector and pick the underside of the canopy. Notice that we could select other bodies to be shelled at the same time. Set a shell thickness of **2**. There is no option to create a new body - that makes sense since we are just modifying an existing one (the same applies to rounds). Accept the shell with *OK*. Pick the canopy and use the *Show All Except* command in the mini-toolbar.

Add another *Shell* feature (thickness **10**). Pick Body 1 for the shell (Does it have to be the default body?) and in the **Remove surfaces** collector pick on the top surface that will be under the canopy. See Figure 29. Perhaps you can see why we kept the arms separate from the body this long.
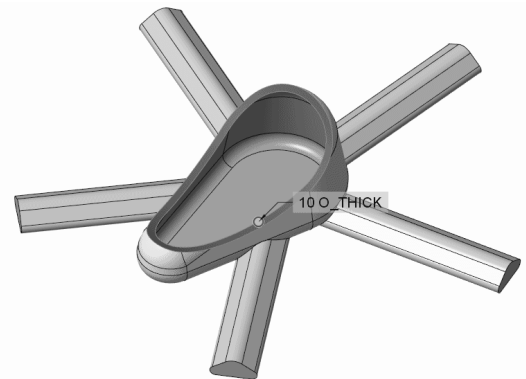
We can now merge the arms with the body:

Boolean Operations ➤ *Merge*

Figure 29 Preview of shell created on Body 1

Pick **Body to modify** as Body 1, and in the **Modifying bodies** select Body 2 that contains all the merged arms. Accept the operation. How do the arms now appear in the

model tree? How many bodies now contain geometry?

Let's add a final solid feature to Body 1. Make it the default body. The feature is a revolve, sketched on Right, with axis A_1 as the axis of revolution. The sketch is a **φ40** circle aligned with the end surface of the arm sticking out the back of the body. This is a 360° revolve. The preview is shown in Figure 30.

We have completed the construction of all the model parts. Have a look at the model tree to see how all the bodies are represented. Where do the shell features appear? Since features can be added to bodies at practically any time, the structure of the model tree can easily get confusing. It is a good idea, therefore, to name bodies and features to help distinguish them. Also, note that the **Design Tree** can be detached from the **Model Tree** into its own floating window, even onto a separate monitor screen if you're lucky enough to have one.
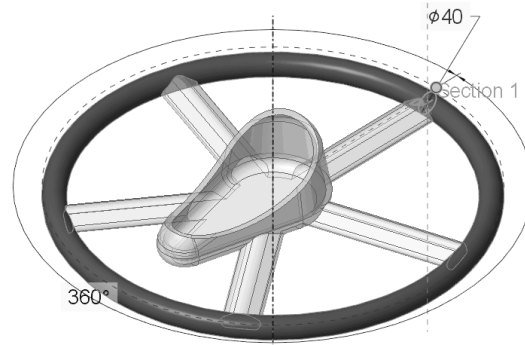
Figure 30 Preview of Revolve added to Body 1

Select Body 1 and in the RMB menu, pick **Rename**. Call the body something like *cockpit*. Select Body 8 and rename it as *canopy*. It's probably a better idea to rename bodies as you go rather than retrospectively as we did here.

By the way, have you saved the file recently?

Creating Parts from Bodies

We may have need of separate part files for the two completed bodies in this part. This is very easy to do. Select the *canopy* body in the model tree, and in the RMB menu select

> **Create Part From Body**

You can then provide a name for the part (might as well call it *canopy*) and indicate if you want to use your default part template (always a good idea). This creates the part and opens it in its own window. Check the model tree for the part. The geometry brought in from the master model file *mbody* appears as an **Extern Copy Geom** feature at the bottom of the model tree. What material has been assigned? Can you change that to something different (like a transparent plastic)? Where do you have to change this?

Repeat this for the *cockpit* body.

Create a new assembly called **podracer** using your default assembly template and **Assemble** the two components (*canopy* and *cockpit*) that are **In Session**. Constrain each of these using the **Default** constraint. This constraint is not required (ie other constraints could be used), but works here since both parts are using the same datums for references.

Now that the parts are in an assembly, they can be ***Exploded*** to get the view in Figure 14 above. Each part can also be analyzed on its own (mass properties or stress analysis using Creo Simulate, for example).

Open the *cockpit* part and try to change the radius of the arc at the front (our very first feature). The ***Edit Dimension*** command is not available for the **Extern Copy Geom** feature. Go back to the master part *mbody*, select the very first extrude in the model tree and change the radius from **50** to **75**. ***Regenerate*** the model and go back to the assembly window. You may have to Regenerate again so that the geometry change can propagate through the master model, the separate part file, and into the assembly model. Voila! The assembly shows the new geometry, shown in Figure 14. Save the assembly file.

There is one potential issue with using a master model part like this if changes to the underlying geometry are likely. Since the models are associative, wherever the component parts go (*canopy* and *cockpit*, in this case), the master part file *mbody* must go as well. If the master file is left behind (in another directory or even another computer!), the **Extern Copy Geom** feature can regenerate but modifications will not be possible. There will also be warnings about missing references. Check it out!

This concludes our brief look at basic multibody functions and operations. You should check out the ***Move***, ***Copy***, ***Mirror***, and ***Split*** operations on your own. Bodies can also be placed in libraries of user defined features (UDFs, see Lesson 5). In some modeling situations, these can be very effective to achieve your modeling goals. Like feature-based modeling, some careful thought and planning is required to determine a modeling plan. You are limited only by your imagination. There are some on-line tutorials that illustrate more complex scenarios where multibody modeling makes things a lot easier. In the ***Help*** area go to

> **Creo Tutorials**
> > **Creo Parametric Tutorials**
> > > **Multibody Tutorials**

Introduction to the Project

The assembly project to be completed in this tutorial involves the modeling and assembly of the three-wheeled utility cart shown in Figure 31. The cart contains 26 or so parts, many of which are repeated in the assembly. The total assembly has about 75 parts (mostly bolts!). We will use the techniques introduced in the lessons to model various parts of the cart as exercises at the end of each lesson. We will average about 4 parts per lesson, so you should get lots of practice! In the final lesson, we will assemble the cart, using a number of advanced functions for dealing with assemblies. Try not to "jump the gun" on this assembly task, since the functions to be covered in the last lesson can really speed up your job of putting the cart together.

For your modeling exercises, the parts shown at the end of each lesson[5] will illustrate the critical dimensions. A figure will also be provided to show where the parts fit into the overall assembly. **Only the critical dimensions are shown on each part - you can use your judgement and creativity to determine the remaining dimensions.** In this regard, take note of the following:

♦   ALL UNITS ARE IN MILLIMETERS! We set up the default part template with this setting.

♦   Dimensions are usually multiples of 5mm. For instance, all the plate material and the wall of the cargo box are 5mm thick. The tubing is 25mm square.

♦   All holes and cylinders, unless otherwise dimensioned, are $\varphi 10$. This applies to bolt holes, pins, rods, and so on.

♦   All holes, unless otherwise dimensioned, are coaxial with cylindrical surfaces or located on symmetry planes.

♦   For some of the trickier parts, in addition to the figures showing the dimensions, there will be some discussion and hints to help you get going.
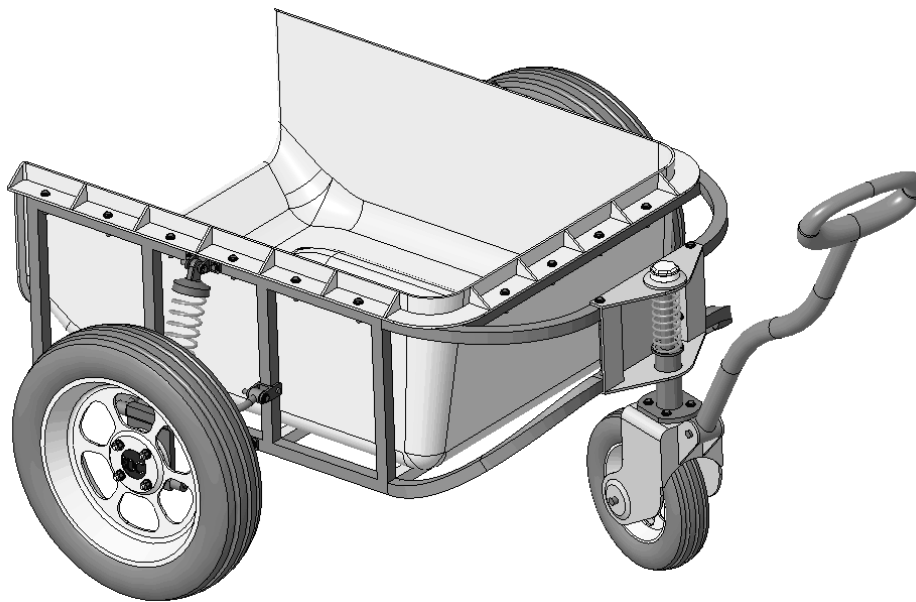

Figure 31 The assembly project - a three-wheeled utility cart

When we get to the final assembly in Lesson 8[6], remember that it is an easy matter to modify dimensions of the various parts so that the assembly fits together. Don't be too concerned when you are modeling the parts if you have to guess at one or two

---

[5] A complete list of project parts is included in the preface (right after the Table of Contents) to this book.

[6] Note that it is possible and maybe a good idea to create your own assembly template. See Exercise #1.

dimensions. These can be modified later if the need arises.

When you are creating the parts, try to be aware of the design intent for the part and how it might eventually be placed in the assembly[7]. For example, if the part has one or more planes of symmetry, it is common practice to use the default datum planes for these. In the assembly, the *Align* constraint using these datum planes is an easy way to position the part (usually with another symmetric part).

Although a suggested part name is given, feel free to make up your own part names (although this might cause confusion in Lesson 8!). Remember that Creo is fussy about files that get renamed in isolation, or moved to another directory. If a part has been used in an assembly (or sub-assembly) or drawing, make sure the assembly or drawing is in session if you rename or move the part so that the related files can also be updated.

Summary

This lesson should have given you enough ideas and ammunition to allow you to customize the interface so that it will be most efficient for the type of work that you do. There are a surprising number of users who are unaware of the many options available in *config.pro*. Check them out!

Multibody modeling adds yet another tool to your modeling toolkit. It will probably take some practice to be able to figure out and plan how to use these functions effectively. Feel free to do that for exercises and projects in this book. It is certainly not necessary to use multibody modeling all the time and, in fact, if overused will just unnecessarily complicate things. The key is to be able to recognize situations where the CSG Boolean operations can best solve your modeling problems.

In the next lesson we will look at functions directly involved in model creation. These are for the creation of sweeps.

Questions for Review

1.   What is the name of the file containing your configuration settings?
2.   How can you change the appearance of the command buttons on ribbons? What are the options?
3.   When, and from where, are your configuration settings loaded? Why is there more than one location?

---

[7] You might like to look ahead to the last lesson to see what assembly constraints are used for each part.

4.   What happens if your configuration file contains multiple entries for the same option, each with different values?
5.   How can you find out where your start-up directory is?
6.   How can you create/edit/delete configuration settings?
7.   When do configuration settings become active?
8.   Is it possible to have more than one customized screen layout?
9.   How do you move the Graphics Toolbar in the graphics window?
10.   How do you add/delete buttons on the toolbars?
11.   How do you create your own ribbon group? What are the restrictions on commands that can be placed there?
12.   How do you add a button to the Quick Access toolbar?
13.   What is a mapkey?
14.   Why do you usually want to keep mapkey names short?
15.   How do you attach a mapkey to a keyboard function key?
16.   How do you create a new mapkey?
17.   Are new mapkeys stored automatically? Where?
18.   What is the purpose of a part template? Where are they stored and how do you access them?
19.   How can you prevent the Browser or Navigator pane from opening automatically when you launch Creo?
20.   Are mapkeys case sensitive?
21.   Does a body have to be the default body to add a round to an edge?
22.   What happens if a tangent edge spans multiple bodies and you create a round on it?
23.   In the *mbody* model, was it necessary to merge the five arms before subtracting the *cockpit* body?
24.   What happens if you merge bodies containing different materials?
25.   In which feature creation dashboards does the **Create new body** option appear? In which does it NOT appear?
26.   Can you shell more than one body at a time? What if they have shared surfaces?

Exercises

1.   Create an assembly template. This should have named datums and named views to match your view selection mapkeys and default units to match your default part template. Make this the default template for assemblies using *config.pro*.
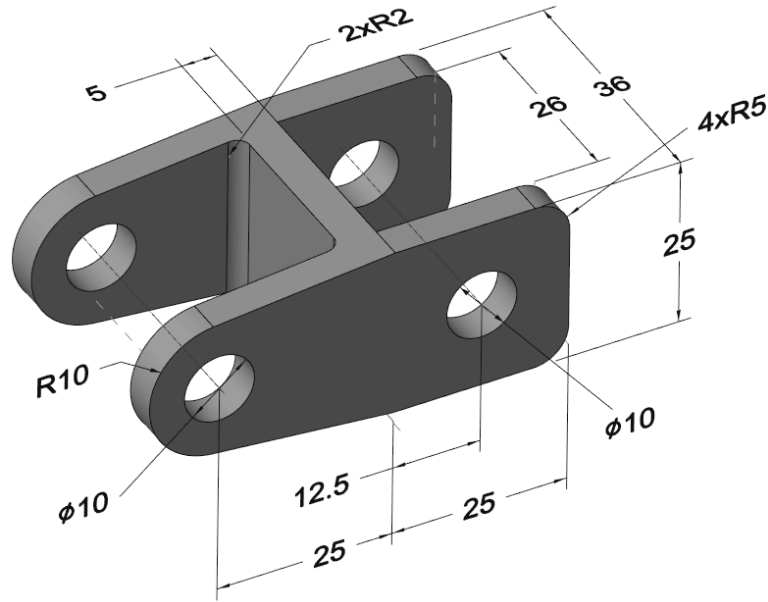
Project Exercises

We're going to start off with some of the easier parts in the cart. These should give you some time to experiment with your configuration file, mapkeys, and part template. The project parts are shown in the figures below. Their location in the cart is also shown here for reference.
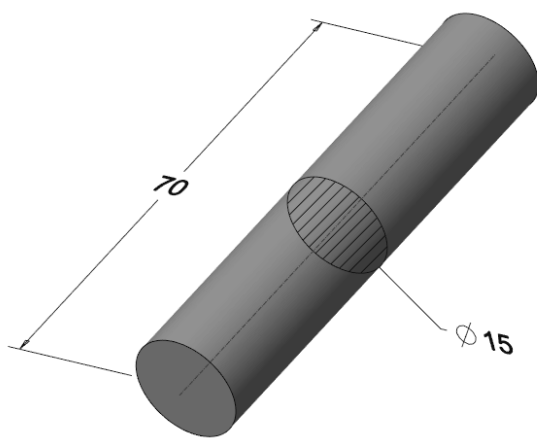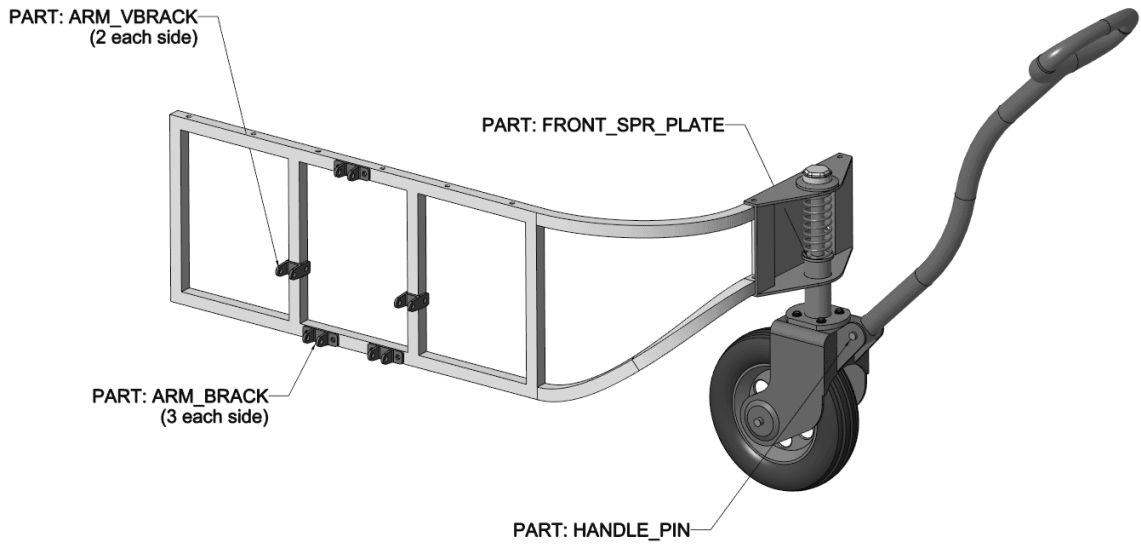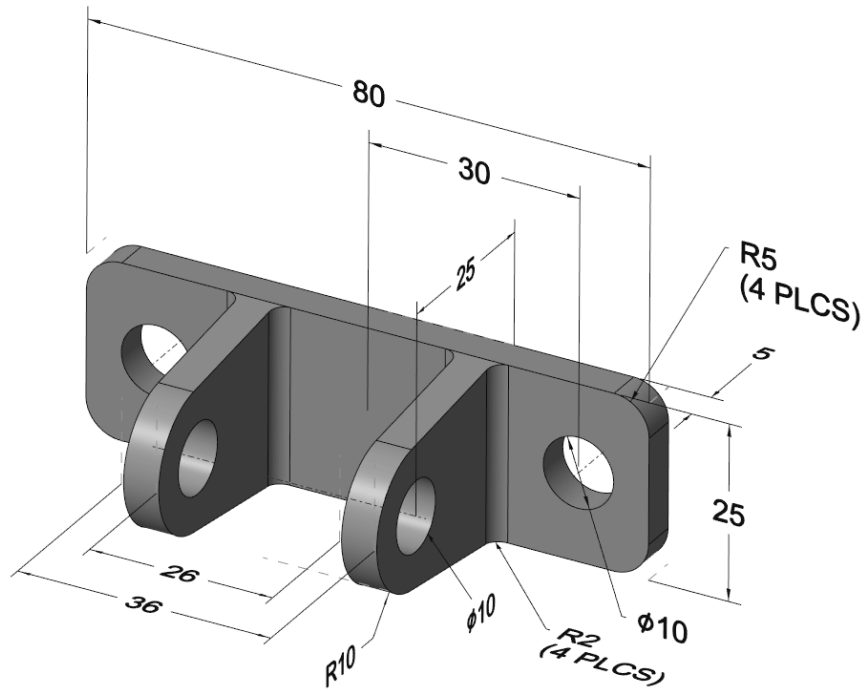

Part: *handle_pin*
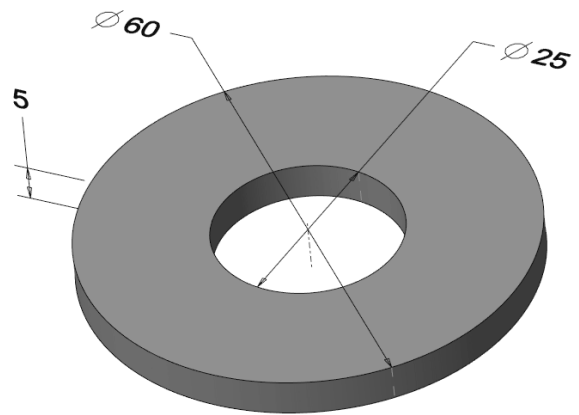
Part: *front_spr_plate*

Part: *arm_vbrack*

Part: *arm_brack*