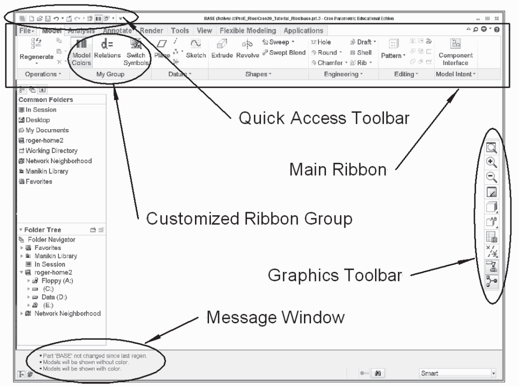
Advanced Tutorial for Creo™ Parametric Releases 1.0 & 2.0

Roger Toogood, Ph.D., P. Eng.

SDC PUBLICATIONS
Schroff Development Corporation



- Part 'BASE' not changed since last regen.
- Models will be shown without color.
- Models will be shown with color.

Lesson 1

Customization Tools and Project Introduction

Synopsis:

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

Overview

This lesson will introduce tools for customizing your configuration and 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 interface so that you can work most comfortably and efficiently. The major customization tool is the use of one or more configuration files (default file *config.pro*). The lesson also includes managing and creating your own custom toolbars and mapkeys. We'll also see how you can create your own part templates. The major project used in this tutorial is introduced and the first four parts are presented.

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.

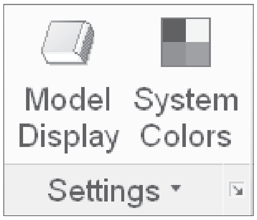
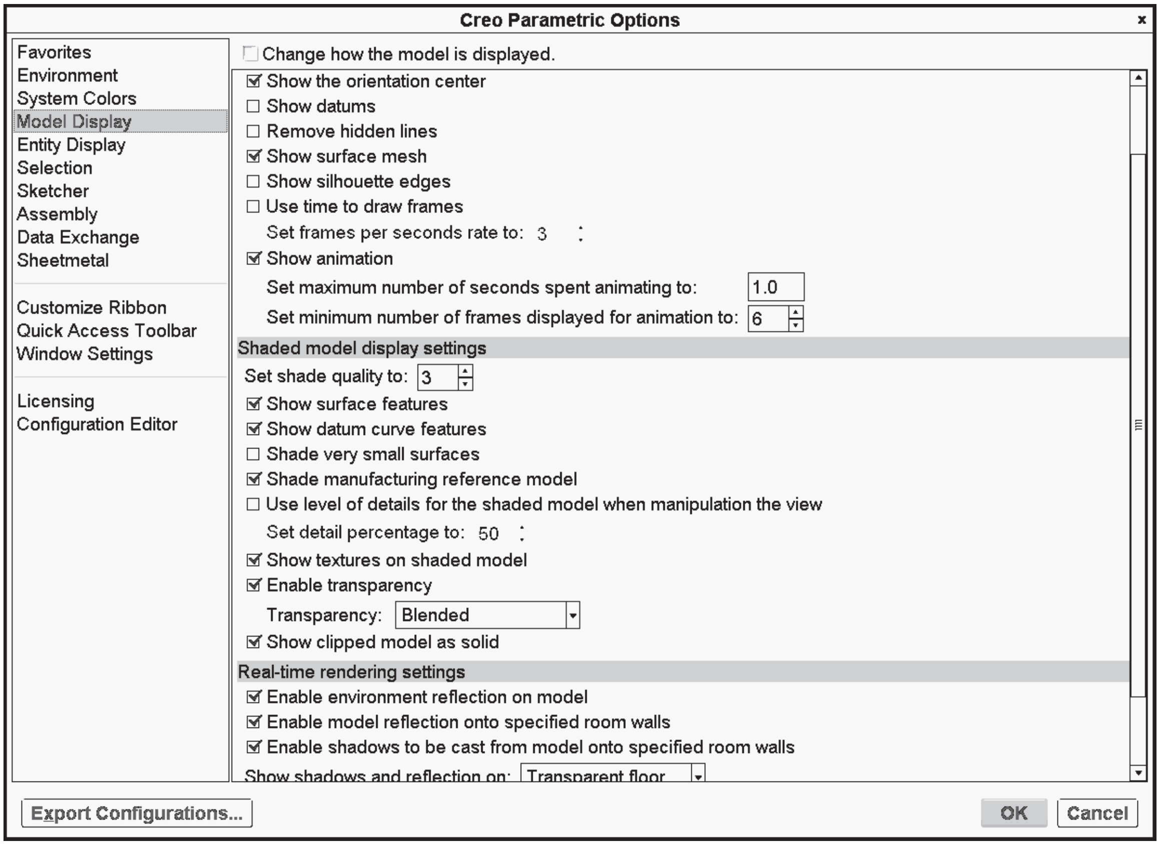


Figure 1 The **Settings** group in the **Home** ribbon

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 Colors***), 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 (ie. when you have a part or assembly loaded), this window is available by selecting ***File ➤ Options***.



□ Change how the model is displayed.

☑ Show the orientation center

□ Show datums

□ Remove hidden lines

☑ Show surface mesh

□ Show silhouette edges

□ Use time to draw frames

Set frames per seconds rate to: 3

☑ Show animation

Set maximum number of seconds spent animating to: 1.0

Set minimum number of frames displayed for animation to: 6

Shaded model display settings

Set shade quality to: 3

☑ Show surface features

☑ Show datum curve features

□ Shade very small surfaces

☑ Shade manufacturing reference model

□ Use level of details for the shaded model when manipulation the view

Set detail percentage to: 50

☑ Show textures on shaded model

☑ Enable transparency

Transparency: Blended

☑ Show clipped model as solid

Real-time rendering settings

☑ Enable environment reflection on model

☑ Enable model reflection onto specified room walls

☑ Enable shadows to be cast from model onto specified room walls

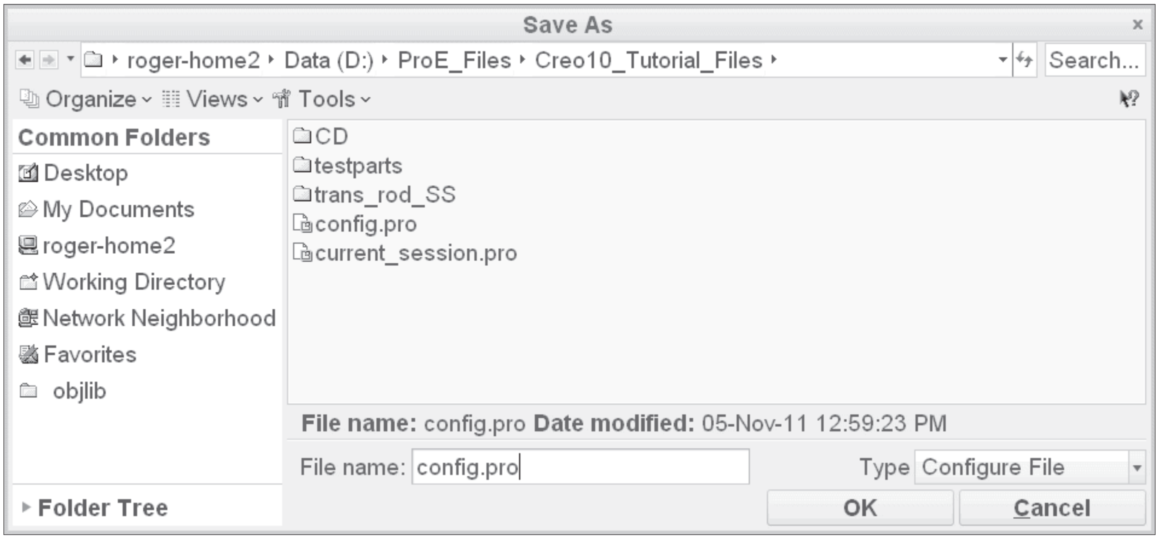
Show shadows and reflection on: Transparent floor

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 and options available.

Any setting changes made will be valid for the current session unless you explicitly save them. That is done using the ***Export Configurations*** button at the bottom of the window. In addition, at the bottom of many of the panels is an ***Import/Export*** button (that also allows you to read in a file of previously created settings).

If you have made settings changes but have not explicitly saved them, when you select ***OK*** in the **Options** window, you will be asked if you want to save the settings. A “yes” will open the dialog window shown in Figure 3. The default name and location for the configuration settings file is *config.pro* in the current working directory. If you want the settings to “stick” (be active the next time you launch Creo), this is exactly where you want them saved. If you don’t want the settings read in at start-up, but do want them available later for a specific project, you can save the configuration file anywhere. It can then be loaded at any time using the ***Import*** function.



File name: config.pro Date modified: 05-Nov-11 12:59:23 PM

File name: config.pro

Type Configure File

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 most important configuration setting file is a special file called *config.pro* that is automatically read from the startup directory 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 users but is not usually available for modification by users. Your system administrator has control of this file.
- Creo loadpoint/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]. This is highlighted in the Navigator on start-up.

[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.

Settings made in the first copy (*config.sup*) 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 part libraries, 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). Be aware that when a new configuration file is read in, some options may not take effect until Creo is restarted. This is discussed more a bit later.

The Configuration File Editor

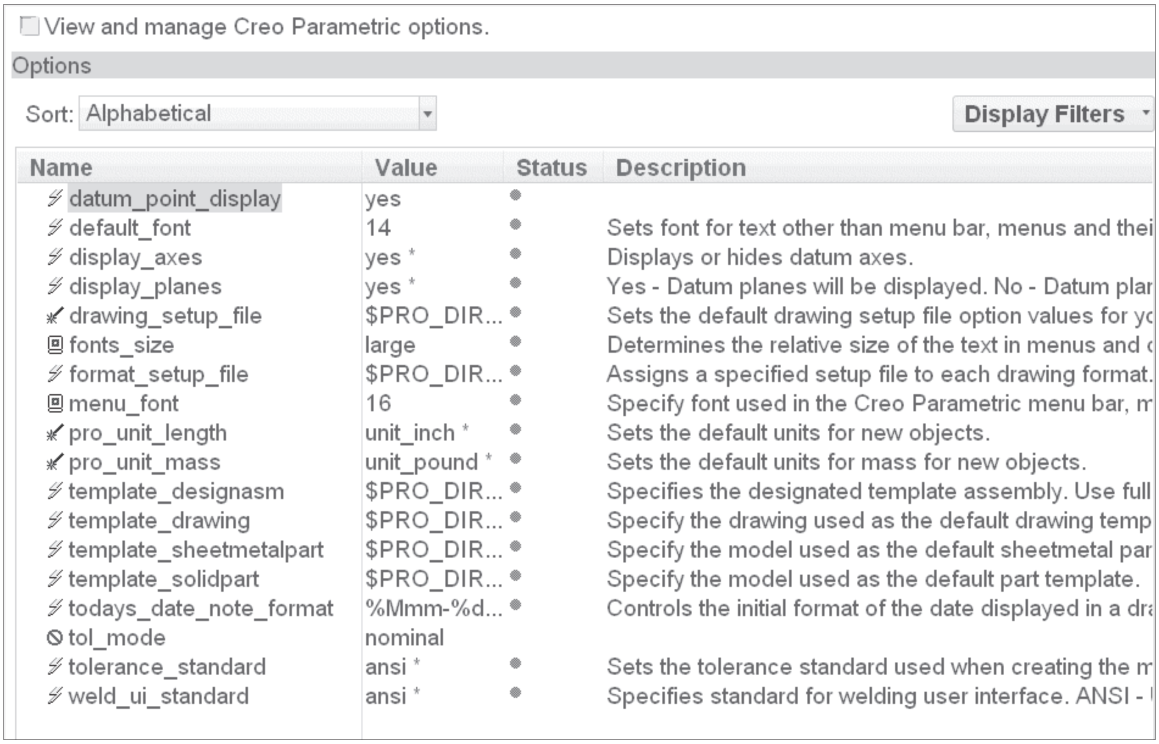
You can access your current configuration file using

File ➤ Options ➤ Configuration Editor

This brings up the **Options** window with the existing *config.pro* contents shown in the right panel. See Figure 4.

In the ***Display Filters*** pull-down at the right, deselect the option to ***Show current session 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 the *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 (about 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. Fortunately, there are a couple of tools to help you find the setting you're looking for. Let's see how they work.



□ View and manage Creo Parametric options.

Options

Sort: Alphabetical

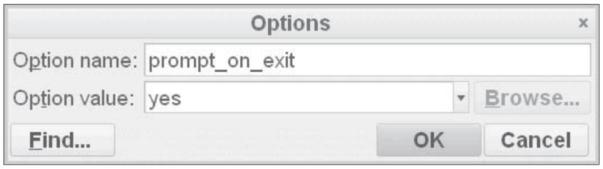
Name	Value	Status	Description
datum_point_display	yes	●	
default_font	14	●	Sets font for text other than menu bar, menus and thei
display_axes	yes *	●	Displays or hides datum axes.
display_planes	yes *	●	Yes - Datum planes will be displayed. No - Datum plar
drawing_setup_file	$PRO_DIR...	●	Sets the default drawing setup file option values for yc
fonts_size	large	●	Determines the relative size of the text in menus and c
format_setup_file	$PRO_DIR...	●	Assigns a specified setup file to each drawing format.
menu_font	16	●	Specify font used in the Creo Parametric menu bar, m
pro_unit_length	unit_inch *	●	Sets the default units for new objects.
pro_unit_mass	unit_pound *	●	Sets the default units for mass for new objects.
template_designasm	$PRO_DIR...	●	Specifies the designated template assembly. Use full
template_drawing	$PRO_DIR...	●	Specify the drawing used as the default drawing temp
template_sheetmetalpart	$PRO_DIR...	●	Specify the model used as the default sheetmetal par
template_solidpart	$PRO_DIR...	●	Specify the model used as the default part template.
todays_date_note_format	%Mmm-%d...	●	Controls the initial format of the date displayed in a dra
tol_mode	nominal		
tolerance_standard	ansi *	●	Sets the tolerance standard used when creating the m
weld_ui_standard	ansi *	●	Specifies standard for welding user interface. ANSI -

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

Open the **Display Filters** again and check the box beside **Show current session options** and select ***Sort(Alphabetical)***.

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.



Options

Option name: prompt_on_exit

Option value: yes

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 **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

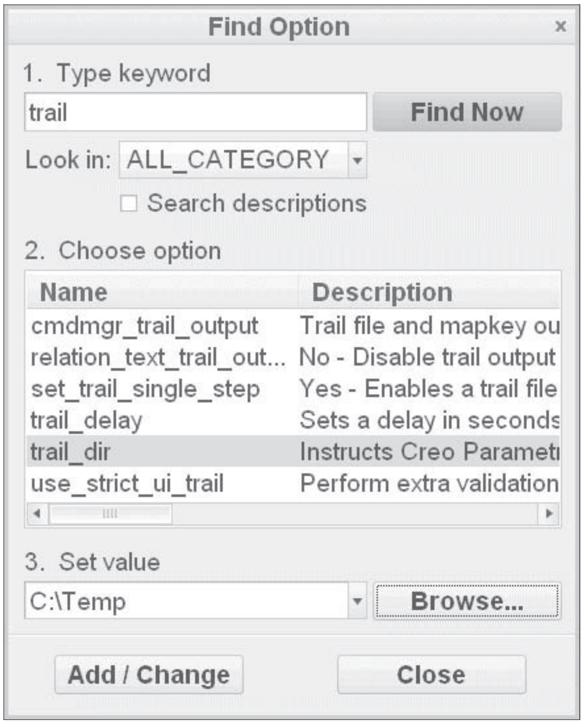
spin_with_part_entities **yes**

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 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 ***c:\temp***. Then select ***Add/Change***. The new entry appears in the **Options** window. In the **Find Option** window, select ***Close***.



trail

Look in: ALL_CATEGORY

☐ Search descriptions

Name	Description
cmdmgr_trail_output	Trail file and mapkey ou
relation_text_trail_out...	No - Disable trail output
set_trail_single_step	Yes - Enables a trail file
trail_delay	Sets a delay in seconds
trail_dir	Instructs Creo Paramet
use_strict_ui_trail	Perform extra validation

C:\Temp

Figure 6 Using ***Find*** to locate an option

For some options, the value is numeric (eg setting a default tolerance, number of digits, or the color of entities on the screen). In these cases, you can enter the relevant number (or numbers separated by either spaces or commas). For example, select ***Add*** and enter the name **system_hidden_color**. Then beside **Value**, enter the numbers **60 60 60** (separated by spaces). These give the values of red, green, and blue (out of 100). Equal values yield gray; this setting will brighten the hidden lines a bit from the default value.

Select ***OK*** or just hit ***Enter***.

We have now specified five options. The green stars in the **Status** column indicate they have not taken effect yet.

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, 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	**1**

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. This symbol also occurs for Creo Simulate options when you are in Creo Parametric.

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.

Saving Your *config.pro* Settings

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, type in the desired name for the file (or accept the default) - in this case ***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***. Note that these settings will be read in but not activated immediately (note the green star). That will happen when you close the window with ***OK***.

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.

Checking Your Configuration Options

Because some settings will not activate until Creo is restarted (mainly dealing with windows properties like fonts and colors), 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 working 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.

Now use the RMB pop-up on the Quick Access toolbar at the top. A pop-up will allow you to specify the

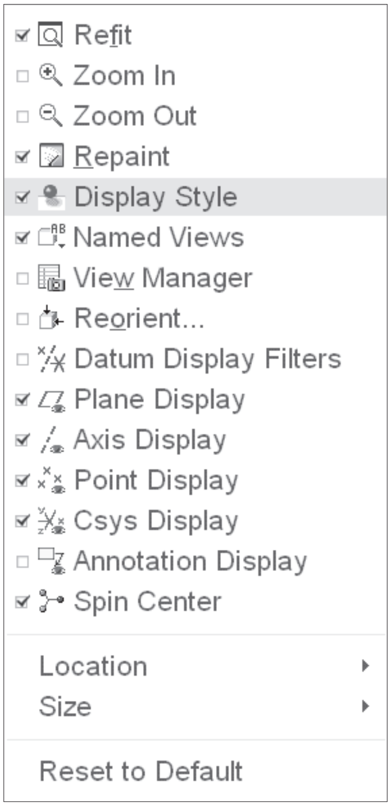


Figure 7 Customizing the **Graphics** toolbar

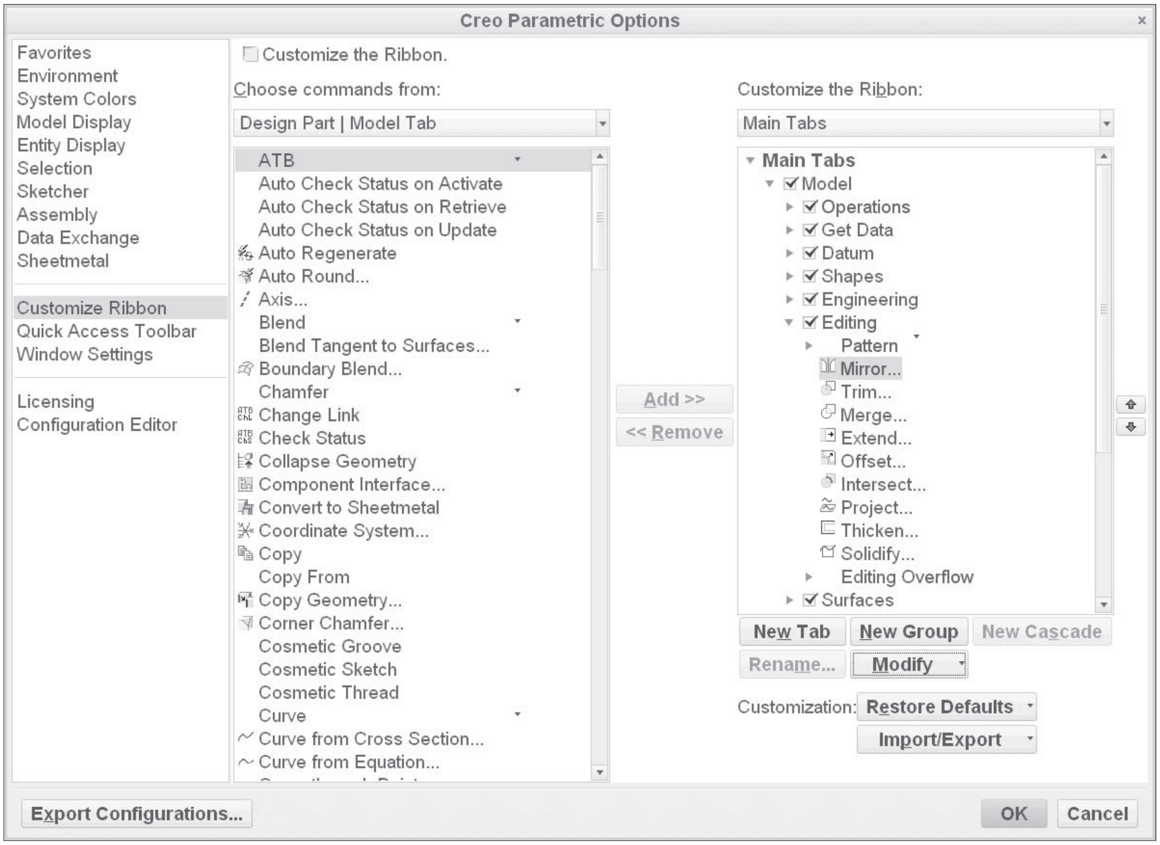
location (relative to the ribbon) or remove it entirely, or modify the commands. If you select

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. As usual, for this to "stick" you must save the settings using ***Export***. 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).

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 left pane, and the ribbon structure is shown in the right pane. Each of the tabs and groups can be expanded to show the commands it contains.



Choose commands from: Design Part | Model Tab

Customize the Ribbon: Main Tabs

Figure 8 Customizing the Creo Parametric ribbons

We cannot modify the command structure of the existing ribbons, tabs, and groups. However, we can do a lot to modify the appearance. 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 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 the ***New Group*** button. 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***.

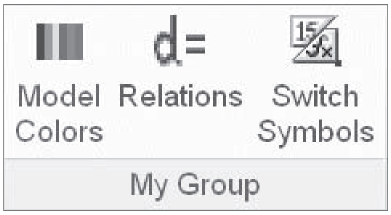


Figure 9 A customized ribbon group

As usual, to make these changes permanent, you must ***Export*** them.

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 to open the following path in the Help Center:

Creo Parametric
- **Fundamentals**
  - **Fundamentals**
    - **Configuring Creo Parametric Fundamentals**

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

Keyboard Shortcuts - Hotkeys

The table below shows most of the default hotkeys. These are shortcuts to commonly used commands. As indicated, 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).

Table 1-1 Creo hotkeys

Key	Command	Key	Command
Ctrl - A	***A***ctivate window	Ctrl - P	***P***rint
Ctrl - C	***C***opy feature	Ctrl - R	***R***epaint graphics window
Ctrl - D	***D***efault view orientation	Ctrl - S	***S***ave file
Ctrl - G	re***G***enerate model	Ctrl - V	paste feature
Ctrl - N	create ***N***ew model	Ctrl - Y	redo
Ctrl - O	***O***pen existing model	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 a 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 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. Mapkeys are meant to be used for extended command sequences that you use frequently.

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. For the same reason, a 3-character mapkey can never have the same two first letters as a 2-character mapkey.

Ideally, you would like to have mapkeys that are very easy to remember, like “vd” (view default), or “rg” (regenerate). 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!).

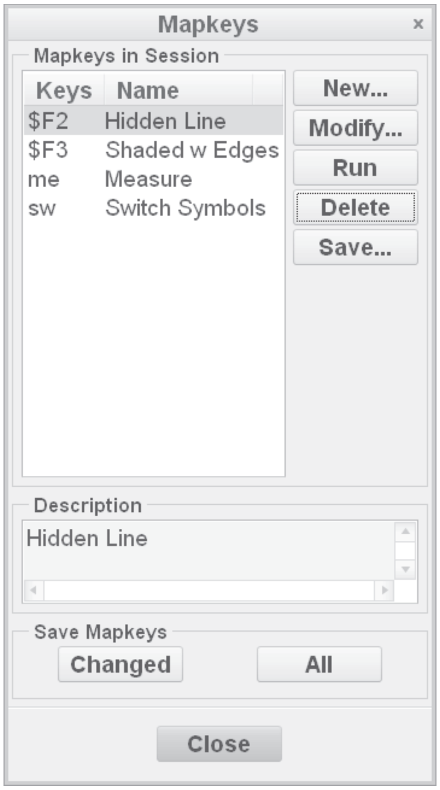
Listing Current Mapkeys

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 above the main graphics window. Mapkeys that start with a “$” are function keys.

Note that 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.



Keys	Name
$F2	Hidden Line
$F3	Shaded w Edges
me	Measure
sw	Switch Symbols

Description: Hidden Line

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), you can modify the keyboard sequence (for example, use “dv” instead of “vd”) for the new mapkey.

Creating Mapkeys

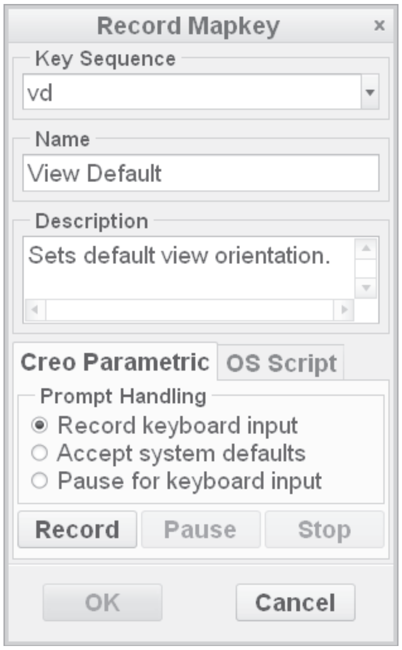
New mapkeys are created as follows. We will create a very simple mapkey sequence "vd" that will reorient the view to the default orientation and, from wherever you are in the ribbons, return you to the **Model** ribbon. To set this up, you will have to bring in one of your previously created parts. Do 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



Key Sequence: vd

Name: View Default

Description: Sets default view orientation.

Prompt Handling
● Record keyboard input
○ Accept system defaults
○ Pause for keyboard input

Figure 11 Creating a mapkey

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 and select the 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. When saving a mapkey you can choose either *config.pro* or *current_session.pro*. There are three ways to save the mapkeys using the buttons in the **Mapkeys** window:

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

Remember that if you save the mapkey in the *current_session.pro* or elsewhere, it will not be loaded automatically the next time you start Creo. To do that, you must explicitly save the mapkey definitions into the *config.pro* file. 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.

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. Prior to Release 2000i^2 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** template. Notice that it has two parameters (MODELED_BY and DESCRIPTION). 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.

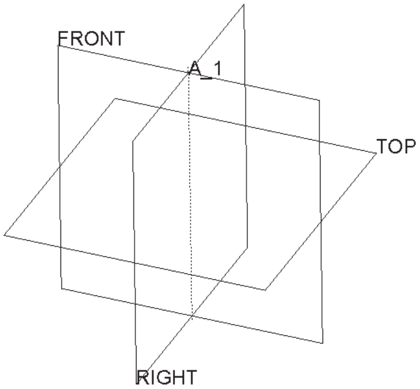


Figure 12 New template file *mytemplate.prt* default datums

Let's create a parameter for the part material. Go to the **Model Intent** group pulldown and select ***Parameters***. The two existing parameters are listed. Click the add button (green "+"), and create a new parameter **MATERIAL** and set the type to ***String***.

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

c:\program files\ptc\creo20\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    $CREO_DIRECTORY\templates\mytemplate.prt
```

When you exit the options window, the new setting should be stored in your default configuration file (*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 checked, enter a name (like ***test***), and select ***OK***. A copy of your custom template (check for the axis and the material parameter) 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 > Customize Ribbon

In the dropdown 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 choose from among several hundred existing icons to replace the happy face with something easier to remember. See Figure 13.

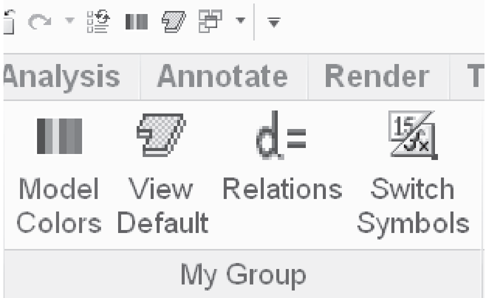


Figure 13 Mapkey added to ribbon group and Quick Access toolbar

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 ***Add***. 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.

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 14. 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[2] 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.

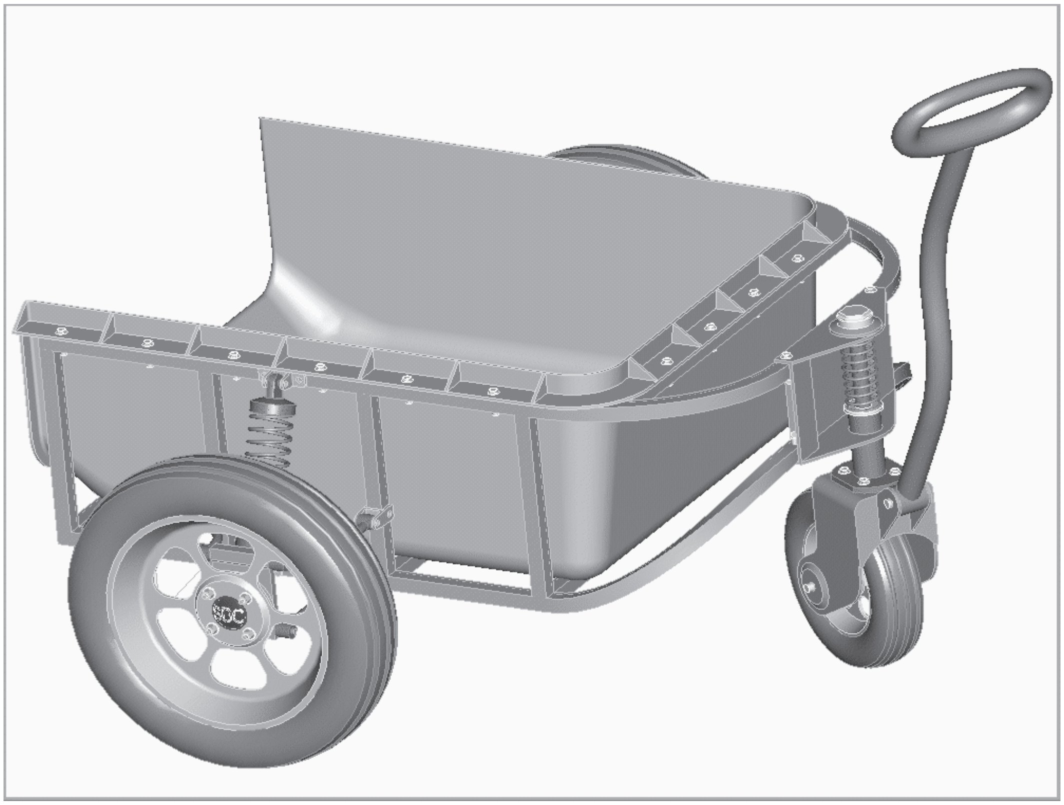


Figure 14 The assembly project - a three-wheeled utility cart

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

When we get to the final assembly in Lesson 8, 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 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[3]. 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!

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?
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?

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

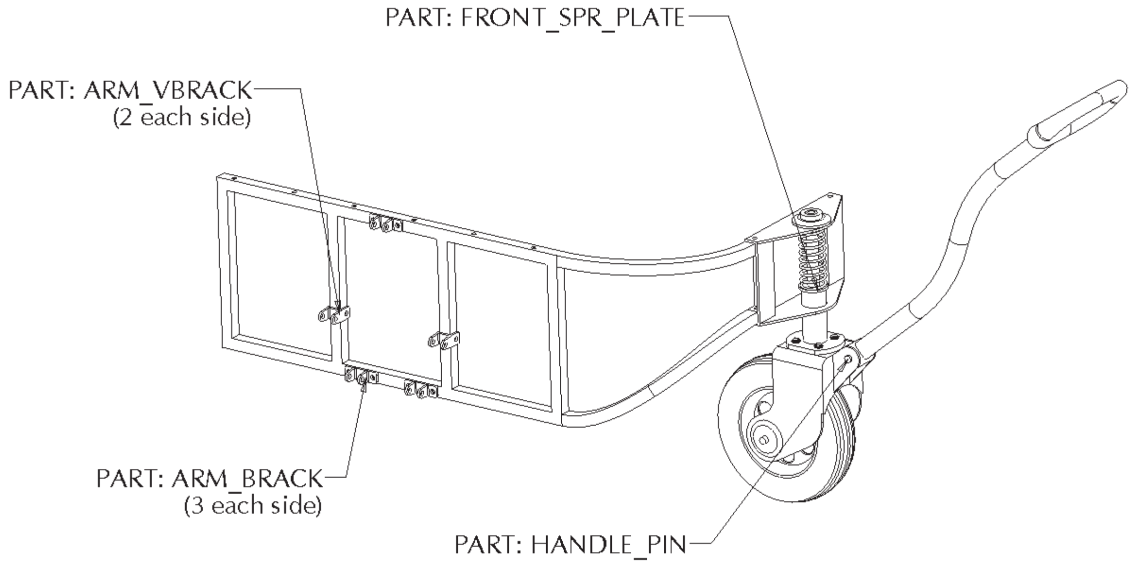
6. How can you create/edit/delete configuration settings?
7. When do configuration settings become active?
8. Is it possible to have more than one customized screen layout?
9. How do you move the Graphics Toolbar in the graphics window?
10. How do you add/delete buttons on the toolbars?
11. How do you 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?

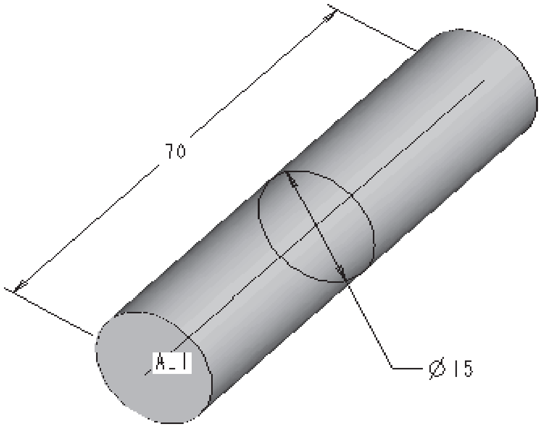
Exercises

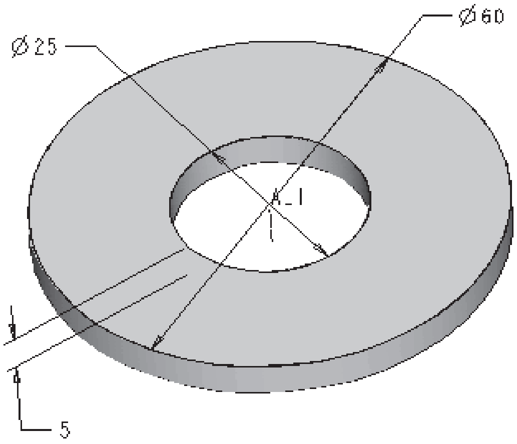
1. Create an assembly template. This should have named datums and named views to match your view selection mapkeys and default units to match your default part template. Make this the default template for assemblies.

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.

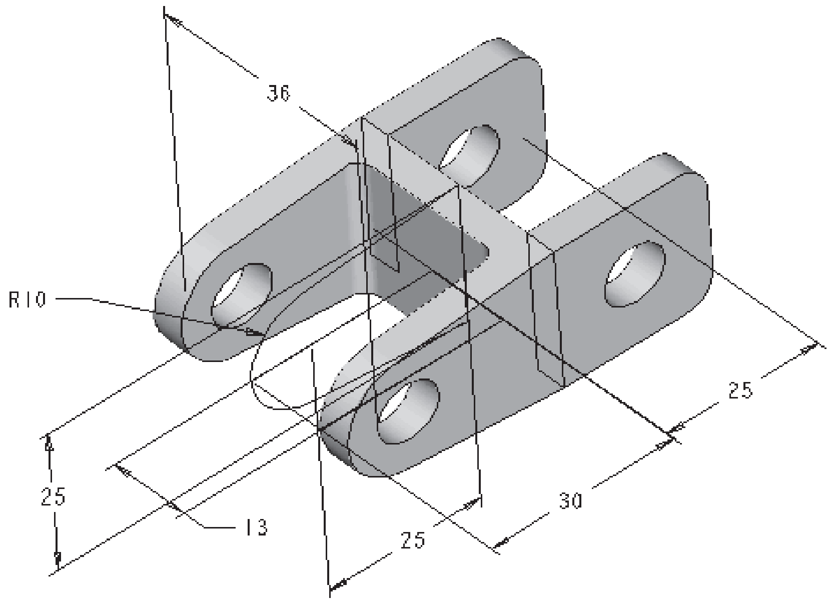




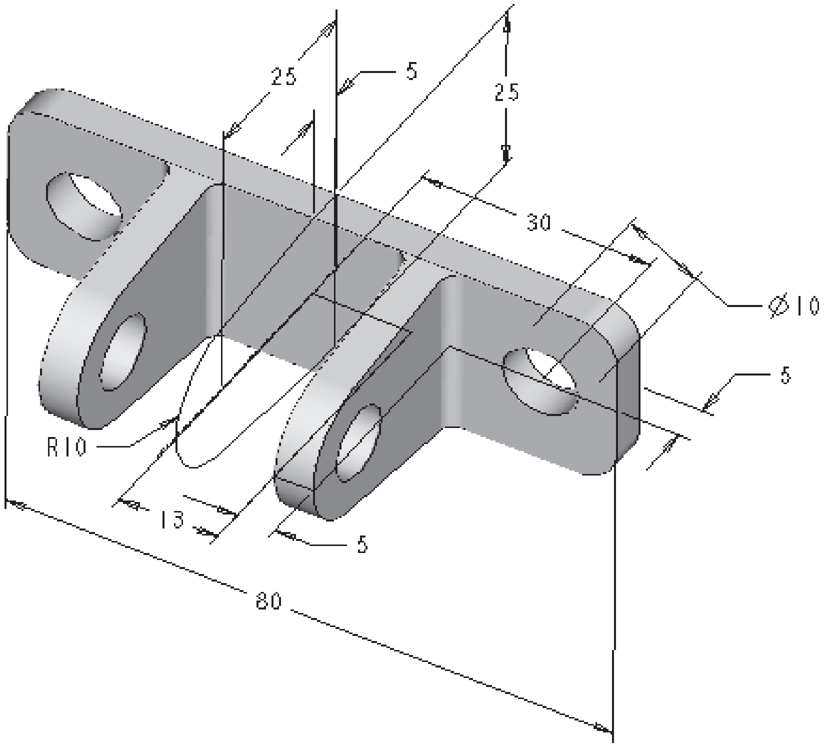


Part: ***handle_pin***

Part: ***front_spr_plate***



Part: ***arm_vbrack***



Part: ***arm_brack***