PROfiles Magazine
Required Reading for the PTC Professional
HOME TIPS & TRICKS PAST ISSUES CO
Spring 2005
Cove

Configuring and Using ModelCHECK with Wildfire 2.0

by Jason Clark, OceanWorks International, Inc.

Before Wildfire and even before Pro/ENGINEER 2001, RAND made this little program called ModelCHECK. After first introducing it as a standalone module, PTC soon acquired the technology and implemented ModelCHECK in its Flexible Engineering Package.

ModelCHECK is now fully integrated into Wildfire 2.0 and takes advantage of the embedded browser window. But even users of earlier builds of Pro/ENGINEER can take advantage of most of the functionality described below. The main things you will be missing are the Pro/INTRALINK integration with using shape indexing and the internal Pro/ENGINEER ModelCHECK configurator (which I rarely use anyway).

Note: This article is written with the Microsoft Windows version of Pro/ENGINEER Wildfire 2.0 in mind. The UNIX and Linux varieties may differ slightly, but the basics are likely to be the same.

First, A Little Legwork

Before configuring ModelCHECK, a little preliminary work is required. For the program to be effective, you have to start by knowing what you want from ModelCHECK.

Current model requirements. Take the time to review your current model requirements and the way you are using parameters, layers, relations and so forth to control and store information. The purpose of reviewing your process is twofold. First, you need to acquire the information you want ModelCHECK to police in order to create consistency. Second, this review provides a good opportunity to identify where you are having issues and how to fix them.

Legacy model requirements. We've all changed model parameters or the layers we use in our models. As such, we've manipulated formats and parameter names to suit our purposes. What ends up happening is we waste time reconfiguring legacy parts to fit within current standards, which may also affect the legacy parts. To streamline the process, we can use ModelCHECK to update old models "live" or revitalize existing models in a batch process.

Planned model requirements. If your review reveals deficiencies or improvements you would like to take advantage of, you can document these changes within ModelCHECK by adding the relevant checks. In this way, existing models can move forward and receive the new changes. Your new requirements can be new parameters, renaming parameters, or creating a layer practice and more!

MORE...
Cover Story

Enhancing Informatio
Exchange

2005 Execu
Meeting Re

Winter TC

Large Custo
Forum

Take Advar
Governmer
Funded
Pro/ENGIN
Training

Reverse-
Engineerin
Fan Blades
American E

Using a Far
Table Insta
Create a Si
Drawing in
Pro/INTRA

Pro/ENGIN
Image Outp
Versus Cop

Using the T
Module of
Pro/MECH
to Optimize
Electrostati
Problem

Inserting 2
Geometry i
Pro/ENGIN

Pro/INTRALINK and ModelCHECK. With Wildfire 2.0, ModelCHECK has become Pro/INTRALINK-aware. While beyond the scope of this article, one of the new advanced features is the ability to use Pro/INTRALINK Commonspace to store information on the topology of your models. ModelCHECK utilizes a shape indexing technology that analytically reviews the model shape, size and feature set and then stores that information called "duplicate data." As ModelCHECK adds to the duplicate data pool, it has the ability to compare models and tell if you are creating a model that already exists. In addition, you can create some fancy check-in rules based on data ModelCHECK writes to the model parameters, allowing Pro/INTRALINK to decide whether an object may be checked in based on ModelCHECK data values.

Key Configuration Areas

Below are key configuration areas to consider when implementing ModelCHECK and configuring ModelCHECK files.

1. Parameters are the most basic and perhaps fastest of checks. Parameters for parts, assemblies and drawings all need to be checked. Also give thought to parameters you've used in the past and would like to update for your current configurations.

2. Layers are essential for managing models. A common practice is to place key features on prescribed layers so that you can use "isolation" layers to allow feature viewing. For example, features that you might place automatically on layers include:

- datum planes
- datum curves
- datum points
- datum axes
- pipe features
- notes.

3. Geometry checks are the most involved because they take the most time to analyze. Bear in mind that these checks are extremely valuable for maintaining good modeling practices. Among other items, you can give errors to:

- buried features
- edges used for references
- geom checks
- family table instance verification status.

For complete information about the wealth of configurable checks you can perform, refer to the ModelCHECK manual in the PDF collection.

ModelCHECK Configuration Files

Configuring ModelCHECK is done by editing the text files, or, if you are really determined, you can use the ModelCHECK configurator within Pro/ENGINEER. I find the configuration tool inadequate for doing large initial setups, although it does have value for single tweaks here and there. The configuration files are found at <wildfire_loadpoint>\modchk\text\usascii\config, with a directory for each style of file (see below).

It is quite common to have multiple configuration files for each category (start, check, etc.). For some users, creating configuration files based on project or product types works best. For those wanting to have checks for multiple product types, I recommend starting with a part and assembly parameter called check_id or some variant. The idea here is to allow the check_id to tell ModelCHECK which configuration files to use. In an ideal scenario, once you have ModelCHECK fully operational and all parts in your collection are updated, you can run the program in an "automatic" mode to select the correct configuration files.

"Start" or .mcs file. The start files are what tell ModelCHECK to interrogate within a Pro/ENGINEER object. For example, if you may want to specify that certain layers or parameters exist, those configurations reside in the start file. A way to distinguish between a start file and a check file is that the start file includes what you would normally configure your start parts to have (parameters, relations, layers, etc.).

Parameter checks

The parameters to set in your .mcs file are PRT_PARAMETER, ASM_PARAMETER, and DRW_PARAMETER. For example,

```
xxx_PARAMETER <parameter_name> <parameter type> EQ <value>

PRT_PARAMETER  PN  NONE  EQ  PDM

PRT_PARAMETER  COMPANY  STR  EQ  Parametric Technology
```

With Pro/INTRALINK, use the setting "PRT_PARAMETER <proi_param_name> STR EQ *", where proi_param_name is:

```
PROI_BRANCH

PROI_MODIFIED

PROI_RELEASE

PROI_REVISION

PROI_VERSION

PDMRL

PDMDB

PDMREV
```

To rename parameters, use the following syntax:

```
PRT_PARAM_RENAME  PART_NUM  PART_NUMBER

ASM_PARAM_RENAME  ASSEM_NME  ASSEMBLY_NAME

DRW_PARAM_RENAME  AUTHOR  DESIGNER
```

Layer checks

Utilizing layer checks is ideal for the pesky datums or layers that are not being blanked when they are supposed to be. This is also extremely helpful to ensure drawings are printing correctly.

The parameters to set in your .mcs file are PRT_LAYER, ASM_LAYER, and DRW_LAYER. For example,

```
xxx_LAYER <layer_name> <layer_status> <feature_type>

PRT_LAYER  AXES  BLANK  DATUM_AXIS

ASM_LAYER  CSYS  BLANK  DATUM_CSYS

DRW_LAYER  CURVES  BLANK  DATUM_CURVES
```

To move layers, you can use the following syntax:

```
PRT_LAYER_MOVE  DTMS  DATUM_PLANES

ASM_LAYER_MOVE  DTMS  DATUM_PLANES

DRW_LAYER_MOVE  DTMS  DATUM_PLANES
```

You can delete layers by first moving them and then rerunning ModelCHECK. After the second ModelCHECK run, you then have the option to delete.

"Check" or .mch file. The check file tells ModelCHECK what checks to use—for example, rename datums, check for early rounds, notify for geom checks, etc. The check file is tab delimited, so it's best to edit in Wordpad. For small changes, the ModelCHECK configurator is also acceptable.

To enable or disable a check, use the following options for each of the categories described below:

Y = Perform the check, if a fail do not report error or warning

N = Do not check

E = Check the item, report as error on fail

W = Check the item, report as warning on fail

The concept behind the check file is to enable the available ModelCHECK options based on "when" to run. ModelCHECK can perform in several modes:

I = Interactive. When ModelCHECK is explicitly run, the check will obey the option noted above.

B = Batch. If you run ModelCHECK in batch mode, the checks will be activated. Using batch is quite useful for converting large quantities of parts or assemblies, and provides a good opportunity to run the more intensive geometry checks.

R = Regeneration. I personally do not use this option as too often you have to close the ModelCHECK window and deal with irate users.

S = Save. This is the best compromise between interactive and regen modes. With the save option, keep it lighter than interactive or again users will be unhappy.

M = Metrics. Using this option, ModelCHECK will report errors to a metrics file. If you use the ModelCHECK Administrator, you can track and view errors reported by users.

"Constant" or .mcn file. The constant file tells ModelCHECK the predetermined values to use for geometry checks. This is typically the lowest deviation value. Lowering the value makes ModelCHECK work harder and therefore take longer to complete its analyses.

"Groups" or .mcg file. This file allows group-based (similar to workgroups, or group permissions) checks within ModelCHECK. To create groups, follow these steps:

1. If one does not exist, create a directory named "groups" in the <ModelCHECK loadpoint>/config directory.

2. In the groups directory, create text files with names of the groups. Each file should have the extension ".mcg".

Example: designers.mcg

3. In the .mcg files, enter the login names of the users that belong to those groups. Specify only one user per line.

```
Example:   jdoe

            njohnson

            ksmith
```

"Text" files. The text files are values for ModelCHECK to use to validate user values for parameters, dimensions, etc. The text files are located in <loadpoint>\modchk\text\usascii\config\text.

Condition.mcc file. The condition file helps ModelCHECK select the proper configurations in an "automatic" mode. It is located in the <loadpoint>\modchk\text\usascii\config folder.

ModelCHECK sequentially evaluates the condition file to utilize the proper configuration. This means that if your model meets more than one condition, it will retain the last successful condition. ModelCHECK currently supports only the "if/else" type of condition. (Wildfire 3.0 may present a ModelCHECK toolkit to provide more options.) The condition "if" is set up as:

```
IF <condition>

IF <condition>

… you may keep adding …
```

```
ELSE

<default condition>

IF (NOCHECK) NOCHECK
```

Conditions are written as one-line statements, so be sure to disable line wrap in your editor. Commented lines appear with a '#' prefix as noted in the example below. Here are some basic examples of the format (each "IF" statement should appear as single lines in your editor):

```
IF (<condition_requirement>) config=(check/<checkfile>.mch)
(start/<startfile>.mcs)
(constant/<constantfile>.mcn)(status/<statusfile>.mcq)

#IF (MODEL EQ fam) config=(check/check1.mch)(start/sample_start.mcs)
(constant/constant1.mcn)(status/sample_status.mcq)

#IF (MODEL_NAME EQ STD_*) config=(check/check1.mch)
(start/sample_start.mcs)
(constant/constant1.mcn)(status/sample_status.mcq)

#IF (EMPTY EQ ??ll) config=(check/check1.mch)(start/sample_start.mcs)
(constant/Constant1.mcn)(status/sample_status.mcq)
```

Note: GROUPNAME can be used in the condition statements to set the config files used by any of the defined groups.

Example:

```
IF (GROUPNAME EQ designers)

config=(check/default_checks.mch)(start/default_start.mcs)
(constant/inch.mcn)
```

config_init.mc file. The configuration initialization file sets the environment for ModelCHECK and is located in the <loadpoint>\modchk\text\usascii\config folder. With the config_init file, you can configure where ModelCHECK stores the report files, duplicate data files, the modes ModelCHECK runs in, and other settings.

The options to set are Y (yes) or N (no) for the interactive (I), Batch (B), Regeneration (R), and Save (S) modes.

About the ModelCHECK Operating Modes

You can run ModelCHECK in four ways, depending on how it is configured.

- **Interactive Mode**—Runs ModelCHECK when you click Analysis > ModelCHECK in Pro/ENGINEER.
- **Regenerate Mode**—Runs ModelCHECK after every regeneration. According to PTC, running ModelCHECK in Regenerate mode is the most effective use of ModelCHECK because running ModelCHECK in this mode, problems are found immediately after they occur. Therefore, errors are easier to correct. While fundamentally correct I do not recommend this option for two reasons. Firstly, it

will irritate the users. Secondly, see the first reason. I only half-joke, because regenerations form a huge percentage of your time in front of Pro/ENGINEER and if you need to address ModelCHECK issues everytime, your users will get weary of using it and curse you for it.

- **Save Mode**—Runs ModelCHECK after every save operation. If you require a more often check I would recommend using this mode instead of the regeneration.
- **Batch Mode**—Runs ModelCHECK outside of Pro/ENGINEER. Use Batch mode to run ModelCHECK on a number of parts, assemblies, or drawings.

Ultimately, running ModelCHECK modes will depend largely on how many users there are and the range of their skill set. If you have seasoned users, running the interactive mode ModelCHECK is probably adequate with some key areas triggered for checking on save (geom checks, buried features etc.). If your users are newer to Pro/ENGINEER perhaps you will want to configure ModelCHECK to run on regeneration mode or save mode so that you catch more errors.

Lastly, the most important thing you can do to offset repeatable ModelCHECK errors is to configure and ensure all users are using start parts. Utilizing start parts will eliminate so many issues that they go hand-in-hand with ModelCHECK as standardization tool.

Configuring ModelCHECK

Here's where the rubber meets the road. Now that you know the files ModelCHECK uses for configuration, it's time to lay out a configuration plan. The following approach is a logical way to proceed, but you can do these steps in any order you want.

- ***Prepare your start check files for any relevant checks***. Keep the checks as broad as you can since ModelCHECK currently supports only three types of custom configs (Light, Medium, Heavy).
- ***Prepare your check files for any relevant check schemes***. Again, keep the checks as broad as possible.
- ***Set up your config_init.mc file.*** Ideally, you want to turn most of the options on and control the checks in your start file as to when they run. In this case, you only need to worry about configuring the start files.
- ***Tweak the constant file.*** I typically leave this as default.

A Comment on ModelCHECK Administrator

For those on Pro/ENGINEER 2001 or earlier, I suggest you look at the ModelCHECK Metrics Program (see TPI 101346, and TPI 101815). I have not yet succeeded in configuring the ModelCHECK Administrator with Wildfire, so hopefully someone from PTC will read this article and find a way to restore the functionality of this program.

A Quick Tutorial

Now we can take what we've gone over and try out a short tutorial. In the example, we will add missing parameters and layers, and rename some old parameters.

For our standard practice, let's say we now require all models contain the following parameters:

PART_NUMBER

PART_NAME

DESCRIPTION

MATERIAL

Let's also assume that our existing parts already have a parameter for part numbers called PART_NO. Examining the required list above, we need a PART_NUMBER parameter. We will use PART_NO and rename it.

For our layer standards, let's use ModelCHECK to place all datum features according to layer names:

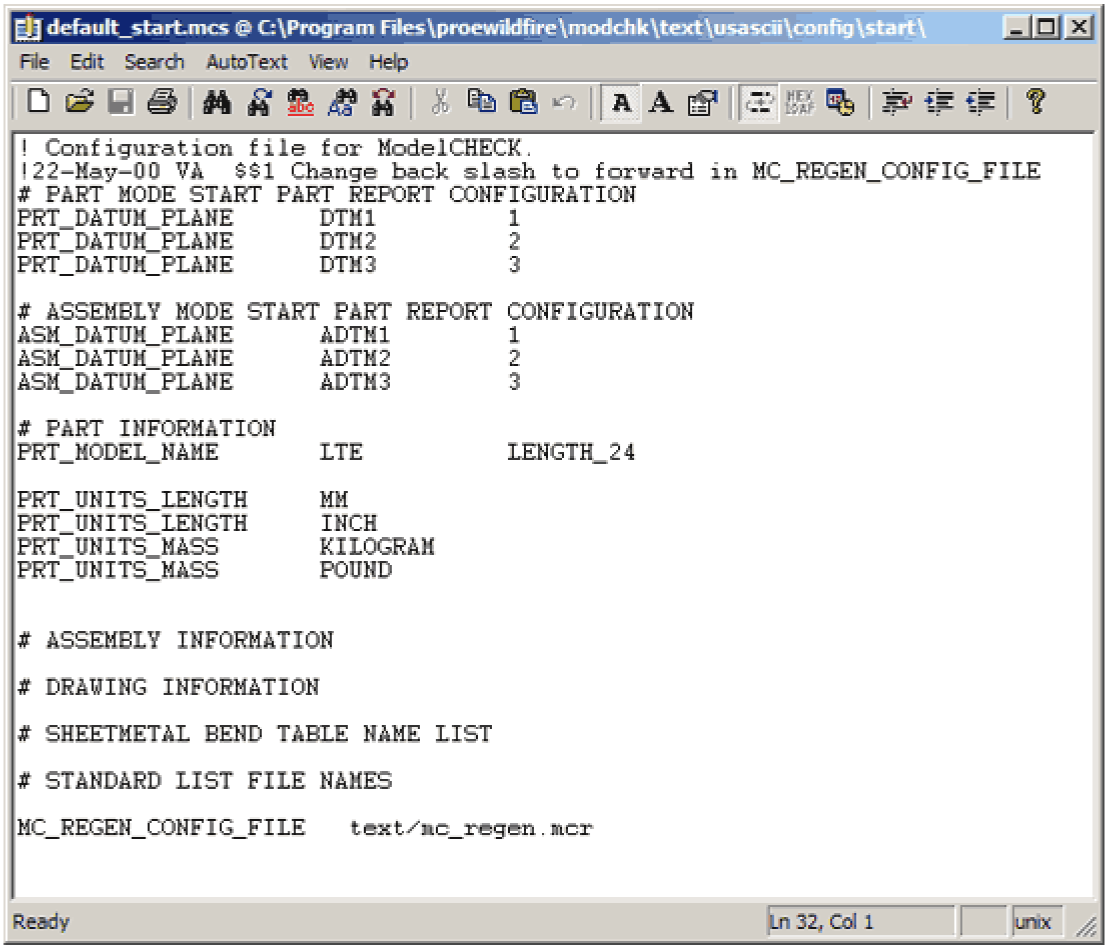
PLANES = all datum planes

AXES = all datum axis features

CSYS = all coordinate system features

For the most part, the default ModelCHECK setup in Pro/ENGINEER is quite easy and useful. For the beginner, I would suggest just tweaking these files to get a sense of editing the files and syntax rather than starting with a blank slate.

1. We will now edit the start check file to incorporate the suggested values. The file we want to open is <loadpoint>\modchk\text\usascii\config\start\default_start.mcs. The default file will most likely appear like so:



default_start.mcs @ C:\Program Files\proewildfire\modchk\text\usascii\config\start\
File Edit Search AutoText View Help
Ready
Ln 32, Col 1
unix

```
! Configuration file for ModelCHECK.
!22-May-00 VA  $$1 Change back slash to forward in MC_REGEN_CONFIG_FILE
# PART MODE START PART REPORT CONFIGURATION
PRT_DATUM_PLANE      DTM1         1
PRT_DATUM_PLANE      DTM2         2
PRT_DATUM_PLANE      DTM3         3

# ASSEMBLY MODE START PART REPORT CONFIGURATION
ASM_DATUM_PLANE      ADTM1        1
ASM_DATUM_PLANE      ADTM2        2
ASM_DATUM_PLANE      ADTM3        3

# PART INFORMATION
PRT_MODEL_NAME       LTE          LENGTH_24

PRT_UNITS_LENGTH     MM
PRT_UNITS_LENGTH     INCH
PRT_UNITS_MASS       KILOGRAM
PRT_UNITS_MASS       POUND


# ASSEMBLY INFORMATION

# DRAWING INFORMATION

# SHEETMETAL BEND TABLE NAME LIST

# STANDARD LIST FILE NAMES

MC_REGEN_CONFIG_FILE   text/mc_regen.mcr
```

2. The section in regards to PRT_DATUM_NAME <datum_name> <number> tells ModelCHECK to see if a part has a datum plane with the name <datum_name> and that it is the feature number <number> in the model tree. I don't normally recommend this because start datums rarely meet any strict order criteria. Instead, delete the lines PRT_DATUM_PLANE and ASM_DATUM_PLANE and enter the following lines for each section:

For the part section:

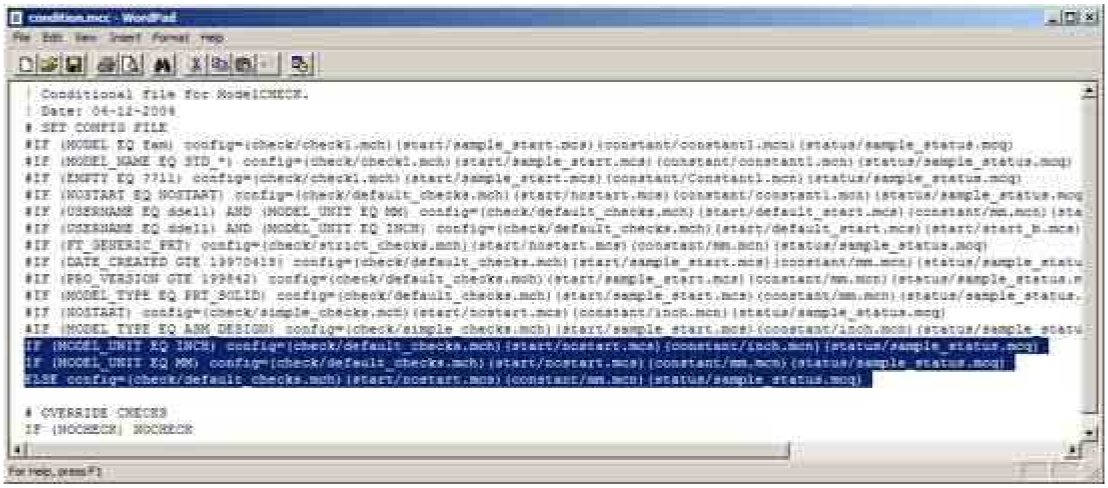
```
PRT_PARAMETER  PART_NAME  STR  EQ  *

PRT_PARAMETER  DESCRIPTION  STR  EQ  *

PRT_PARAMETER  MATERIAL  STR  EQ  *

PRT_PARAM_RENAME  PART_NO  PART_NUMBER

PRT_LAYER   AXES  BLANK  DATUM_AXIS

PRT_LAYER   CSYS  BLANK  DATUM_CSYS

PRT_LAYER   PLANES  BLANK  DATUM_PLANES
```

For the assembly section:

```
ASM_PARAMETER  PART_NAME  STR  EQ  *

ASM_PARAMETER  DESCRIPTION  STR  EQ  *

ASM_PARAMETER  MATERIAL  STR  EQ  *

ASM_PARAM_RENAME  PART_NO  PART_NUMBER

ASM_LAYER   AXES  BLANK  DATUM_AXIS

ASM_LAYER   CSYS  BLANK  DATUM_CSYS

ASM_LAYER   PLANES  BLANK  DATUM_PLANES
```

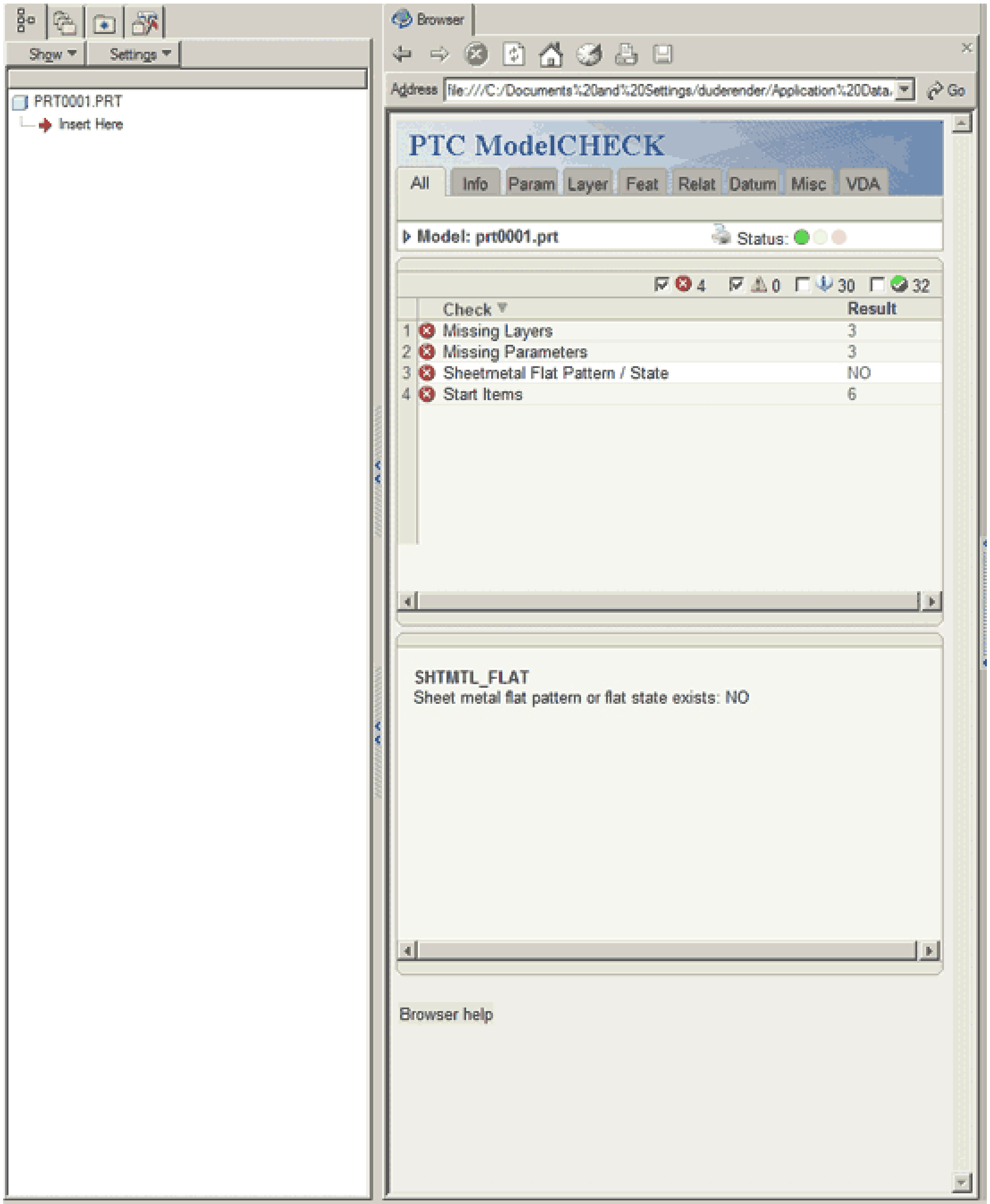
The simple setup here checks for required parameters. If they don't exist, you enter them. The layers are configured to be blanked and contain certain features. A word of insight regarding dual status arrangements like assigning layers with a visibility: ModelCHECK can only change or add one configuration at a time. If a layer doesn't exist and is required, you need to run ModelCHECK twice to create the layer and then again to set the status.

3. Now save the default_start.mcs file. To utilize the settings of the start file we are editing, we need to edit the condition.mcc file located in <loadpoint>\modchk\text\usascii\config. Open the condition.mcc file and edit the highlighted lines shown so that start/nostart.mcs) is replaced with (start/default_start.mcs).



```
! Conditional file for ModelCHECK.
! Date: 04-12-2004
# SET CONFIG FILE
#IF (MODEL EQ fam) config=(check/check1.mch)(start/sample_start.mcs)(constant/constant1.mcn)(status/sample_status.mcq)
#IF (MODEL_NAME EQ STD_*) config=(check/check1.mch)(start/sample_start.mcs)(constant/constant1.mcn)(status/sample_status.mcq)
#IF (EMPTY EQ ?711) config=(check/check1.mch)(start/sample_start.mcs)(constant/Constant1.mcn)(status/sample_status.mcq)
#IF (NOSTART EQ NOSTART) config=(check/default_checks.mch)(start/nostart.mcs)(constant/constant1.mcn)(status/sample_status.mcq
#IF (USERNAME EQ ddell) AND (MODEL_UNIT EQ MM) config=(check/default_checks.mch)(start/default_start.mcs)(constant/mm.mcn)(sta
#IF (USERNAME EQ ddell) AND (MODEL_UNIT EQ INCH) config=(check/default_checks.mch)(start/default_start.mcs)(start/start_b.mcs)
#IF (PT_GENERIC_PRT) config=(check/strict_checks.mch)(start/nostart.mcs)(constant/mm.mcn)(status/sample_status.mcq)
#IF (DATE_CREATED GTE 19970415) config=(check/default_checks.mch)(start/sample_start.mcs)(constant/mm.mcn)(status/sample_statu
#IF (PRO_VERSION GTE 199842) config=(check/default_checks.mch)(start/sample_start.mcs)(constant/mm.mcn)(status/sample_status.m
#IF (MODEL_TYPE EQ PRT_SOLID) config=(check/default_checks.mch)(start/sample_start.mcs)(constant/mm.mcn)(status/sample_status.
#IF (NOSTART) config=(check/simple_checks.mch)(start/nostart.mcs)(constant/inch.mcn)(status/sample_status.mcq)
#IF (MODEL_TYPE EQ ASM_DESIGN) config=(check/simple_checks.mch)(start/sample_start.mcs)(constant/inch.mcn)(status/sample_statu
IF (MODEL_UNIT EQ INCH) config=(check/default_checks.mch)(start/nostart.mcs)(constant/inch.mcn)(status/sample_status.mcq)
IF (MODEL_UNIT EQ MM) config=(check/default_checks.mch)(start/nostart.mcs)(constant/mm.mcn)(status/sample_status.mcq)
ELSE config=(check/default_checks.mch)(start/nostart.mcs)(constant/mm.mcn)(status/sample_status.mcq)

# OVERRIDE CHECKS
IF (NOCHECK) NOCHECK
```

4. Now you can open Pro/ENGINEER and create an empty part with no parameters or layers. With nothing created, let's run ModelCHECK to see what happens. To run ModelCHECK, select **Analysis > ModelCHECK > Interactive**.

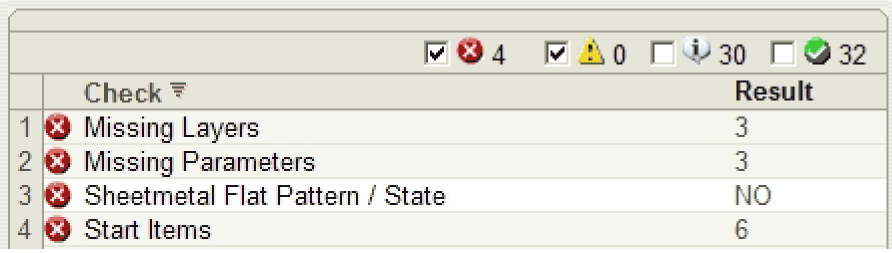


☑ ⊗ 4 ☑ ⚠ 0 □ ℹ 30 □ ✓ 32

	Check	Result
1	⊗ Missing Layers	3
2	⊗ Missing Parameters	3
3	⊗ Sheetmetal Flat Pattern / State	NO
4	⊗ Start Items	6

SHTMTL_FLAT
Sheet metal flat pattern or flat state exists: NO

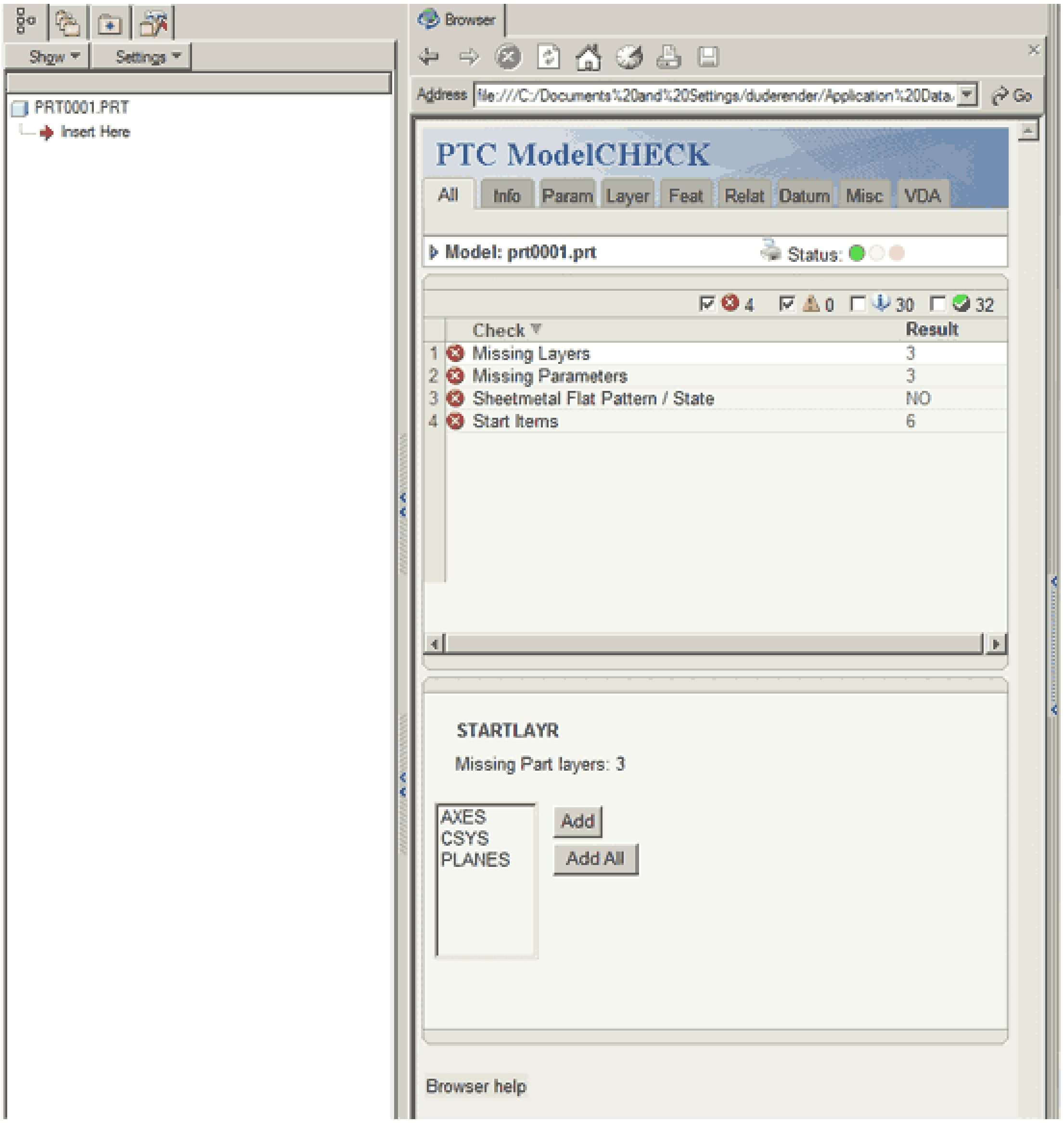
From the interface we see we have warnings and errors. Referring to the picture below the red circles are errors and the yellow triangles are warnings. Typically errors should be configured where you *have to have* something set properly. Suggestions are layer status and required parameters.



☑ ⊗ 4 ☑ ⚠ 0 □ ℹ 30 □ ✓ 32

	Check	Result
1	⊗ Missing Layers	3
2	⊗ Missing Parameters	3
3	⊗ Sheetmetal Flat Pattern / State	NO
4	⊗ Start Items	6

5. Click on the 'Missing Layers' text. This will hyperlink to the errors relating to missing layers:



Address: file:///C:/Documents%20and%20Settings/duderender/Application%20Data

PTC ModelCHECK

Model: prt0001.prt Status:

☑ 4 ☑ 0 □ 30 □ 32

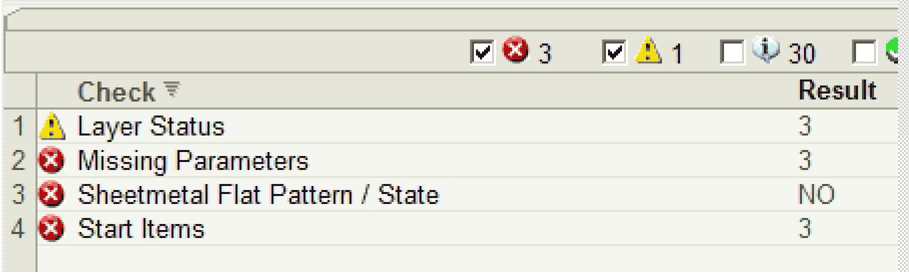
	Check	Result
1	Missing Layers	3
2	Missing Parameters	3
3	Sheetmetal Flat Pattern / State	NO
4	Start Items	6

STARTLAYR

Missing Part layers: 3

AXES
CSYS
PLANES

6. Again we will use the 'Add All' button and rerun ModelCHECK to reflect the layer status warning:



☑ 3 ☑ 1 □ 30 □

	Check	Result
1	Layer Status	3
2	Missing Parameters	3
3	Sheetmetal Flat Pattern / State	NO
4	Start Items	3

7. Now click the 'Layer Status' text and use the 'Update All' button on the linked page.

Last Thoughts

What I've presented within this article only scratches the surface of what you can do with ModelCHECK. Spending the time to configure ModelCHECK pays big dividends in the end by saving time and worry about the consistency of the information users are storing in the models. ModelCHECK is your enforcer of standard practices and really assists administrators who deal with not only Pro/ENGINEER but also PDM or ERP systems that depend on data pushed from Pro/ENGINEER onto drawings or other databases. ♦

Jason Clark is a senior designer and PTC applications administrator at OceanWorks International, Inc. He also chairs the Vancouver Regional User Group. Jason can be reached by email at jclark@oceanworks.cc.