



ARL-SR-0376 • JUNE 2017



Converting Geometry from Creo Parametric to BRL-CAD

by Clifford W Yapp

Approved for public release; distribution is unlimited.

NOTICES

Disclaimers

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

Citation of manufacturer's or trade names does not constitute an official endorsement or approval of the use thereof.

Destroy this report when it is no longer needed. Do not return it to the originator.



Converting Geometry from Creo Parametric to BRL-CAD

by Clifford W Yapp

Survivability/Lethality Analysis Directorate, ARL

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing the burden, to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE (DD-MM-YYYY) June 2017		2. REPORT TYPE Special Report		3. DATES COVERED (From - To) December 2016–May 2017	
4. TITLE AND SUBTITLE Converting Geometry from Creo Parametric to BRL-CAD				5a. CONTRACT NUMBER	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S) Clifford W Yapp				5d. PROJECT NUMBER	
				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) US Army Research Laboratory ATTN: RDRL-SLB-S Aberdeen Proving Ground, MD 21005-5068				8. PERFORMING ORGANIZATION REPORT NUMBER ARL-SR-0376	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)				10. SPONSOR/MONITOR'S ACRONYM(S)	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.					
13. SUPPLEMENTARY NOTES					
14. ABSTRACT There exists in vulnerability/lethality analysis an ongoing need to convert target geometries from commercial systems used by vendors to the BRL-CAD representation used by the US Army Research Laboratory's analysis tools. To make that conversion process more rapid, a plugin was developed for PTC's Pro/ENGINEER computer-aided design system in the early 2000s that simplified conversion of Pro/ENGINEER geometry to BRL-CAD. In 2017, that original plugin is no longer compatible with the newer releases of PTC's tools. This report documents updates and improvements made to the original code base in order to support PTC's Creo Parametric 3.					
15. SUBJECT TERMS geometry, conversion, BRL-CAD, Creo, plugin, parametric					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT UU	18. NUMBER OF PAGES 46	19a. NAME OF RESPONSIBLE PERSON Clifford W Yapp
a. REPORT Unclassified	b. ABSTRACT Unclassified	c. THIS PAGE Unclassified			19b. TELEPHONE NUMBER (Include area code) 410-278-1382

Contents

List of Figures	iv
1. Using Commercially Produced Models in Vulnerability/Lethality Analysis	1
2. Installing the Creo to BRL-CAD Converter	2
3. Converting a Creo Parametric Model	2
4. Other Features of the Creo to BRL-CAD Converter	14
5. Developers Notes: Working with Creo Parametric	19
6. Future Work	33
7. References	35
List of Symbols, Abbreviations, and Acronyms	36
Distribution List	37

List of Figures

Fig. 1	Start up Creo Parametric 3	3
Fig. 2	Select the desired file to open. If you have a large model, you are probably looking for an assembly file. Part files can be converted, but they do not encode multiobject hierarchies.	3
Fig. 3	You may see a “hairball” of informational labels that can be so thick that it hides the actual model shape. Fortunately, the labels can also be hidden as described in Figs. 4–7.	4
Fig. 4	Disable Annotation Display	4
Fig. 5	Select the Datum Display Filters drop-down menu next to the Annotation Display	5
Fig. 6	Disable all of the available categories	5
Fig. 7	Screenshot after disabling categories; ready to load the conversion plugin	6
Fig. 8	From the Tools tab, select Auxiliary Applications	6
Fig. 9	Register plugin to let Creo Parametric know it exists.....	7
Fig. 10	In the File Selection dialog, navigate to the PTC hierarchy Common Files/protokit directory. Select the file creo-brl.dat. This file will be in the directory if the plugin was installed correctly.....	7
Fig. 11	Once registered, the plugin can now be started	8
Fig. 12	Screenshot of successful startup: running report from the Auxiliary Applications dialog, a new “TOOLKIT” group in the tool bar, and a message reporting success in the text status window	8
Fig. 13	Select Tree Columns in the Model Tree view to display data in Creo identifying significant metadata on objects for preservation.....	9
Fig. 14	Various types of parameters are available. As an example, select Model Params for the type of data of interest.....	9
Fig. 15	From the available parameters, select those of possible interest for display in the tree view	10
Fig. 16	Inspect the model to identify data are in at least some of the objects. 10	
Fig. 17	Construct 2 comma-separated lists of attributes using a text editor. The first indicates the parameters to use when constructing objects names. The second lists parameters to preserve as attributes on the corresponding BRL-CAD objects.....	11
Fig. 18	Select File/Creo-BRL to run the converter	11

Fig. 19	The converter dialog offers a variety of settings that impact the conversion process. Populate the various data entry boxes in the lists of Creo model parameters. Select All for verbose logging. (This is not necessary, but is sometimes useful when debugging problems.).....	12
Fig. 20	Once the desired settings are selected, select Go to run the conversion. As the conversion is running, messages should be displayed in the status bar. (This can sometimes fail if the conversion hits an interface update blocking operation, but generally, it should provide a useful sense of how the conversion is progressing.).....	12
Fig. 21	Once the conversion is complete, select Dismiss	13
Fig. 22	Inspect the resulting .g file to ensure the result is as expected	13
Fig. 23	Starting identifier number	14
Fig. 24	Tessellation settings	15
Fig. 25	Disable use of CSG in output.....	15
Fig. 26	Write surface normals when outputting triangle meshes	16
Fig. 27	Reject nonsolid object tessellations	17
Fig. 28	Use bounding boxes to represent objects if mesh generation fails	18
Fig. 29	Creo Parametric provides the ability to suppress features on solids...	19
Fig. 30	When you reach the Application Selection stage of Creo Parametric installation, locate the Customize Applications button.....	20
Fig. 31	Under the Command Configuration tab, locate the PTC Creo Parametric TOOLKIT and ensure it is enabled for installation. If you wish to have Creo Parametric load the TOOLKIT license at startup (you do if you are doing any major development) update the configuration in the Command Configuration tab. See the Creo documentation for more details.	21
Fig. 32	Visual Studio version and toolset are both set in the initial CMake screen	22
Fig. 33	The first time Configure is run, the process will halt with an error. This is expected behavior, because the locations of BRL-CAD and Creo Parametric have not yet been specified. (Both are required for a complete configuration process to succeed.)	23
Fig. 34	For BRLCAD_ROOT_DIR and CREO_ROOT_DIR, select the correct directories to identify the versions of Creo Parametric and BRL-CAD being used to build the plugin.....	24
Fig. 35	Run Configure again, and the process should complete. Select Generate to produce a Visual Studio project. Be sure you have read and understood the messages generated during the configuration process—they are intended to alert you to situations that may result in a failure of the converter plugin to successfully run, or otherwise compromise effective software development work on the converter. 25	25

Fig. 36	Select the generated file “CREO2BRL” to open the Visual Studio project	26
Fig. 37	The target list should include creo-brl, which is the primary target to build. Although it is not required for building, opening the converter source code files in Visual Studio is a good preparation step if you expect to set breakpoints or step through the sources during debugging.....	27
Fig. 38	Building the target should succeed and produce creo-brl.dll. If not, debugging work starts now.	28
Fig. 39	Before the plugin can be loaded, even for testing, it is necessary to run the INSTALL target to make sure all of the required files are in their correct locations in the Creo Parametric directories	29
Fig. 40	If Creo Parametric and the plugin are now running and ready for debugging, then attach Visual Studio’s debugger to the running process.....	30
Fig. 41	From the list of processes, select the one that shows either Creo Parametric (if the plugin is not yet running) or Creo to BRL-CAD (if the plugin is running) and select Attach. Once this is complete, the running plugin will respect break points defined in Visual Studio and the conversion process can be debugged.	31
Fig. 42	Once the plugin is deemed ready for distribution, make sure the build type is set to Release, build the creo-brl target to make sure everything is up to date, and then run the UNLOCK target. This will lock up a Pro/TOOLKIT license for 15 min, but also results in a DLL that can be used by other Creo Parametric installations.....	32
Fig. 43	Once the DLL is unlocked, the PACKAGE target may be used to generate a .zip file and a Nullsoft Scriptable Install System (NSIS) installation executable for distribution.....	33

1. Using Commercially Produced Models in Vulnerability/Lethality Analysis

Vulnerability/lethality (V/L) analysis fundamentally relies on the ability to define for a computer the 3-D space occupied by a target (tank, truck, helicopter, etc.) and the ability to calculate geometric intersection along shot lines through those targets. Historically, models were purpose-built for analysis needs¹; however, preexisting geometric models created by vehicle vendors have been more frequently used over the last 15+ years. Using a preexisting model offers the possibility of a faster and less expensive analysis process, but in order to make use of these preexisting models, they must be converted to the BRL-CAD .g format used by the US Army Research Laboratory's (ARL's) V/L analysis tools (particularly MUVES).²

Current practice is to convert commercial geometries, which typically use Non-Uniform Rational B-Spline (NURBS) as their fundamental geometric shape representation method, to triangle mesh-based geometry. There are many potential paths for such conversions; one simple possibility involves converting all shapes in the commercial system to STereoLithography (STL) meshes and reassembling them manually in BRL-CAD. Unfortunately, while simple, such methods are not only labor intensive but lose all hierarchy and metadata information contained in the original commercial system.

In the early 2000s, John Anderson,³ ARL, wrote a plugin for the PTC Pro/ENGINEER (Pro/E) commercial computer-aided design modeling system that converted triangle meshes generated by Pro/E directly to the BRL-CAD .asc ASCII file format, generated combinations based on the Pro/E geometry hierarchy, and also preserved some metadata information (such as object color). This proved to be a successful conversion strategy, and the original converter has been used ever since to convert commercial models. In 2008 Robert Parker,⁴ ARL, extended this converter to be more flexible about tolerances during model tessellation (the process that generates triangles from the original NURBS shapes) and added a few other behavioral improvements. Since that work was completed, there has been little change to the converter's code or design.

Although the converter code was stable, the PTC products have evolved over time. Pro/E has been replaced by Creo Parametric⁵ and the original Pro/Develop toolkit used for (part of) the original Pro/E to BRL-CAD plugin has been removed in Creo Parametric 3.⁶ This means that existing Pro/E and Creo Parametric version 2 converter binaries will no longer work, and the code will have to be updated to more modern standards before target modelers at ARL can handle vendor-supplied models created in newer Creo versions.

Fortunately, much of the original converter was written using Pro/TOOLKIT rather than Pro/Develop and therefore it requires only minimal updating. However, years of use have also resulted in a number of feature requests to improve the work flows of target modelers utilizing the converter. As the code needed to be modernized in any case, some of the simpler requests have also been targeted for implementation:

- Writing geometry directly to BRL-CAD's binary .g file, instead of staging through the ASCII .asc file format.
- Allowing users to specify model parameters ("attributes" in BRL-CAD terminology) that they wish to have preserved in the conversion.
- Allowing users to specify attributes to be used for generating "user-friendly" BRL-CAD object names to be used in lieu of the Creo part numbers.

All of these goals were achieved, and in the process, the original converter code was extensively refactored and updated.

2. Installing the Creo to BRL-CAD Converter

Installing the converter is simply a matter of placing the correct files in the correct locations within the PTC Creo file hierarchy. The .zip file will decompress the files to the correct relative positions if the proper root level directory is chosen. For example, if you are using Creo Parametric 3 M100 in a standard C:\Program Files installation location, decompress the .zip file in the directory C:\Program Files\PTC\Creo3.0. This may require administrator privileges, depending on how Creo was originally installed.

3. Converting a Creo Parametric Model

When the converter is supplied the name of an existing BRL-CAD file as an output target and that file is the output of a previous conversion from Creo Parametric 3 via this converter, the converter will create and update only those objects that are either newly added or changed since the last conversion. If this is undesirable behavior, either the conversion should target a new .g output file or the modeler should remove specific objects they wish to reconvert.

All BRL-CAD objects are stored using millimeters, so Creo objects will be converted to use those units on a per-part basis.

Figures 1 through 22 illustrate the step-by-step model conversion process, using the open source Pro/E model for the Neo1973 phone.⁷

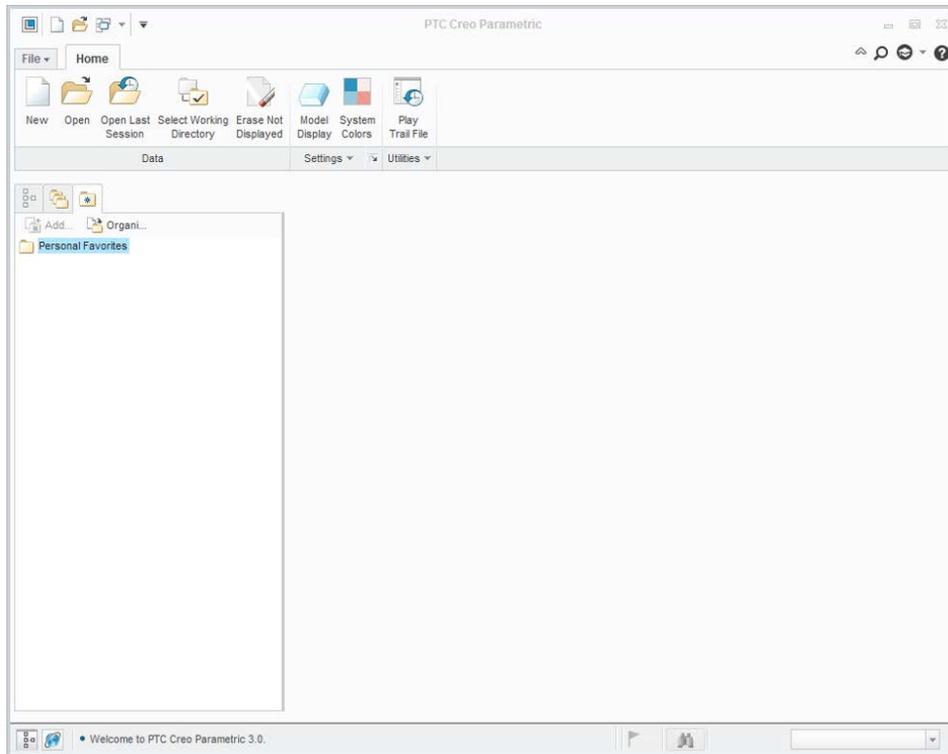


Fig. 1 Start up Creo Parametric 3

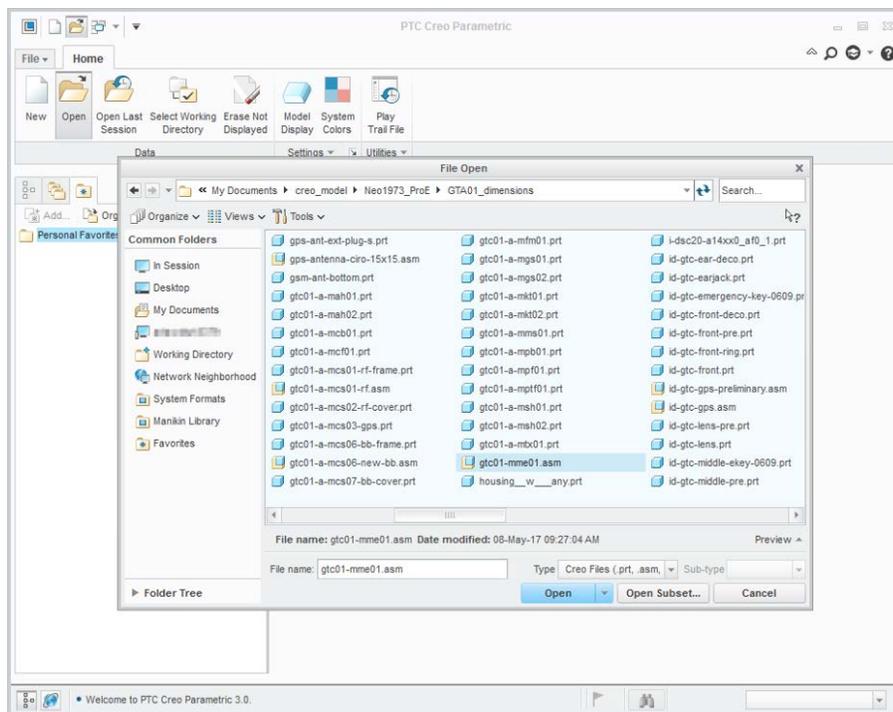


Fig. 2 Select the desired file to open. If you have a large model, you are probably looking for an assembly file. Part files can be converted, but they do not encode multiobject hierarchies.

Approved for public release; distribution is unlimited.

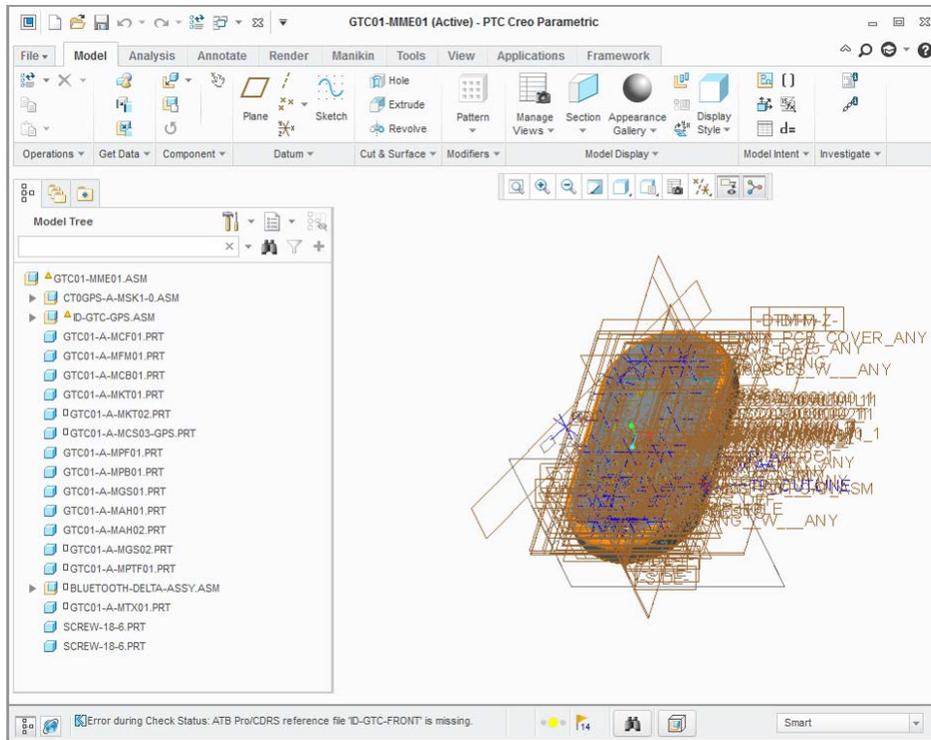


Fig. 3 You may see a “hairball” of informational labels that can be so thick that it hides the actual model shape. Fortunately, the labels can also be hidden as described in Figs. 4–7.

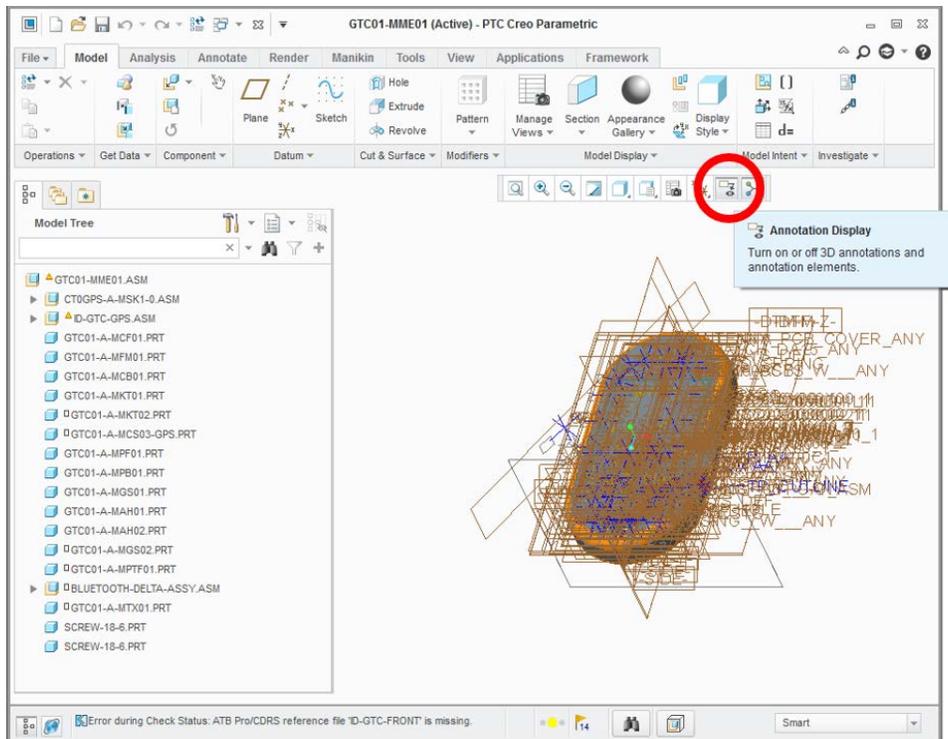


Fig. 4 Disable Annotation Display

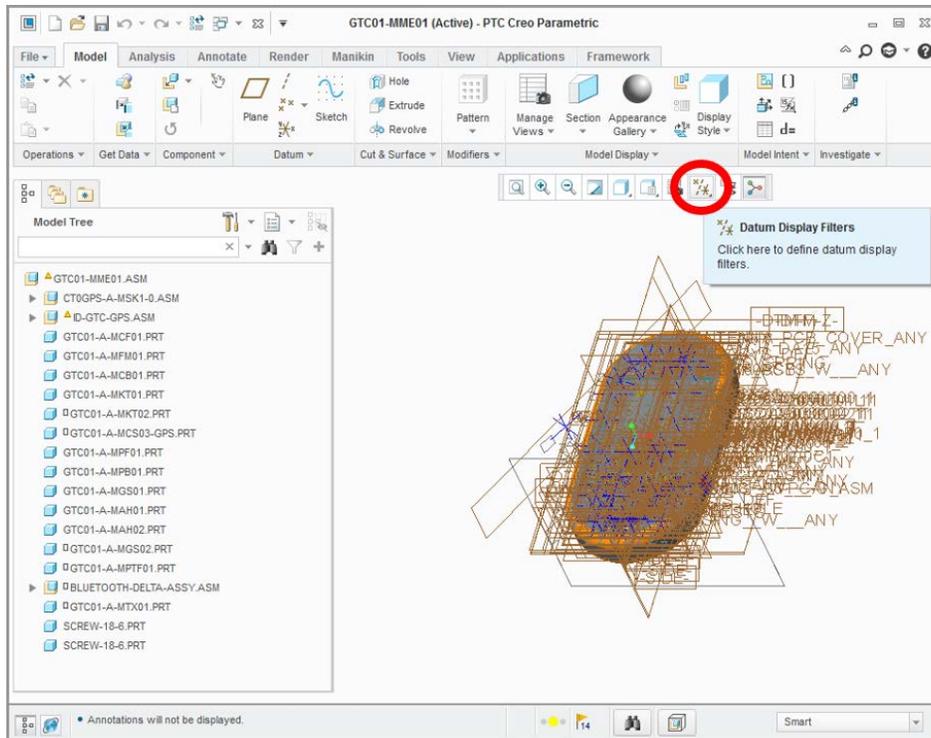


Fig. 5 Select the Datum Display Filters drop-down menu next to the Annotation Display

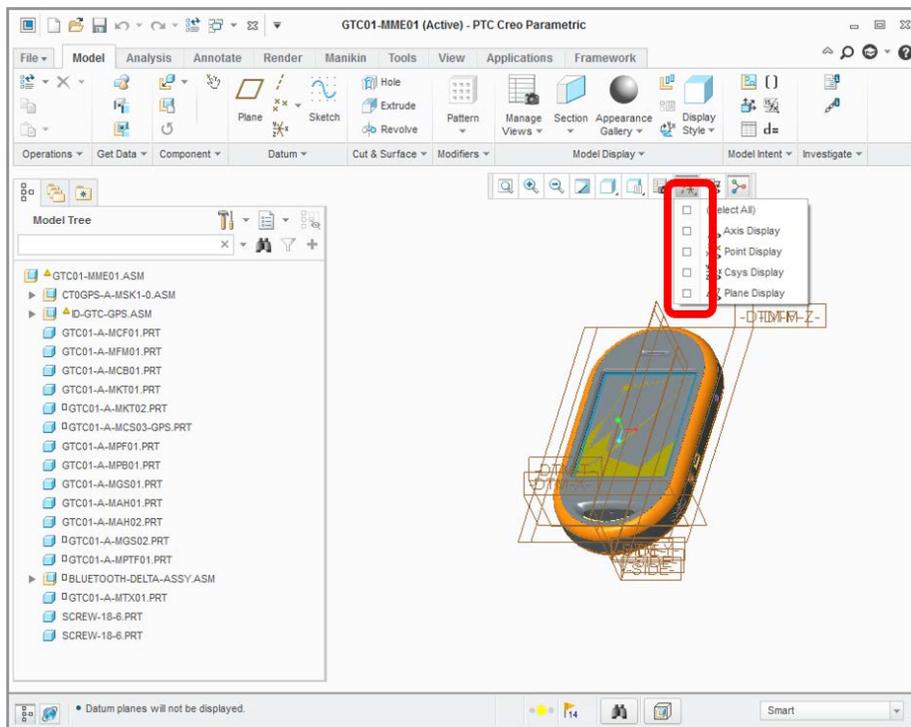


Fig. 6 Disable all of the available categories

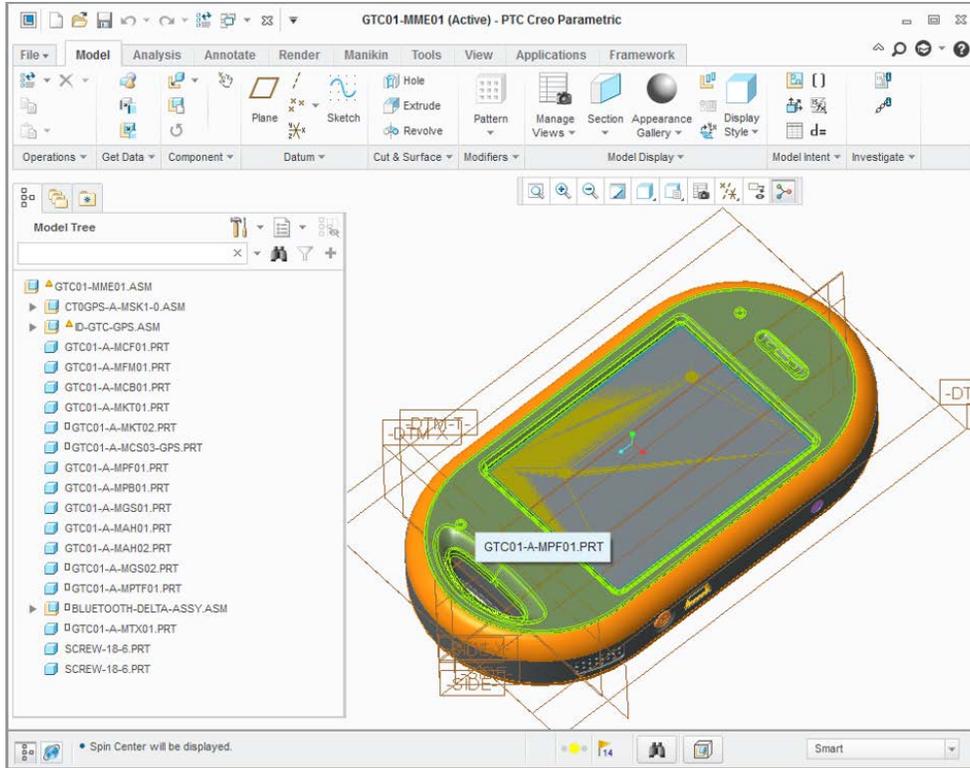


Fig. 7 Screenshot after disabling categories; ready to load the conversion plugin

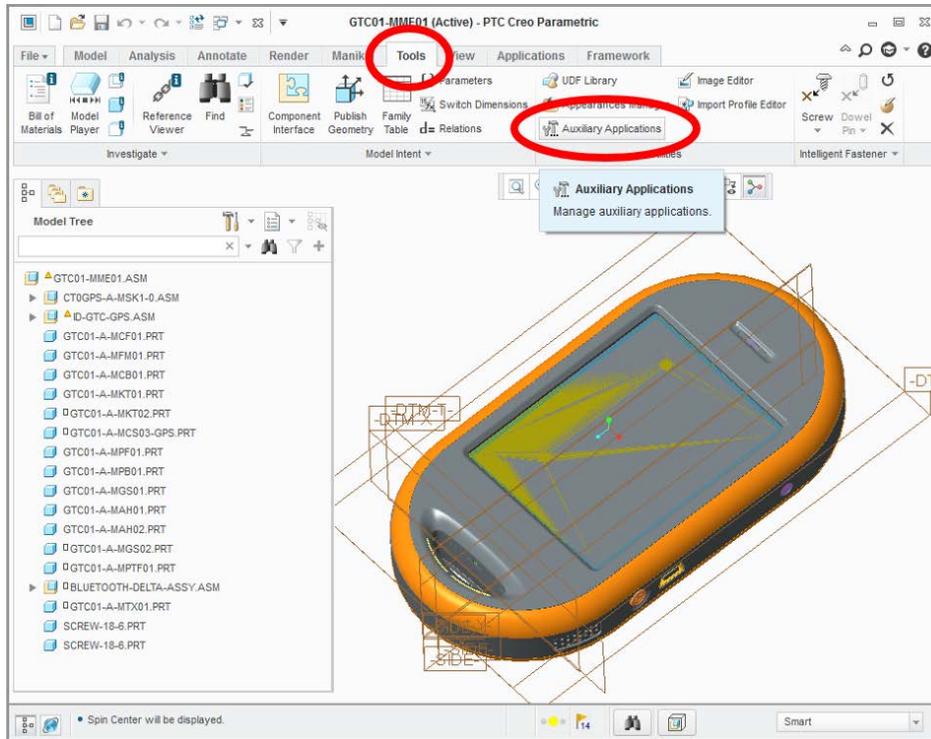


Fig. 8 From the Tools tab, select Auxiliary Applications

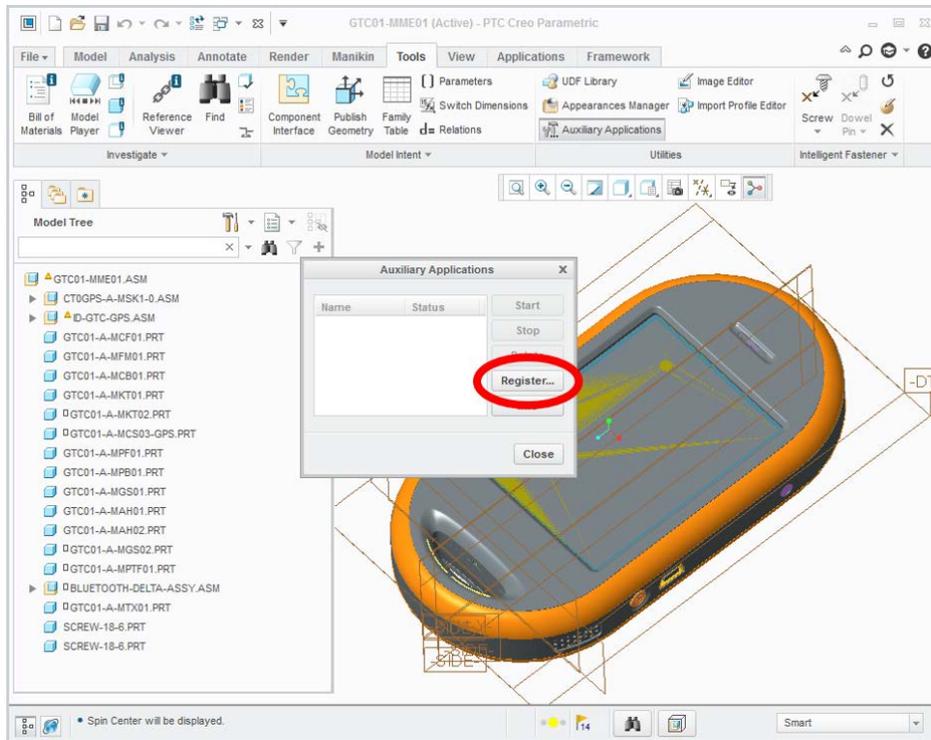


Fig. 9 Register plugin to let Creo Parametric know it exists

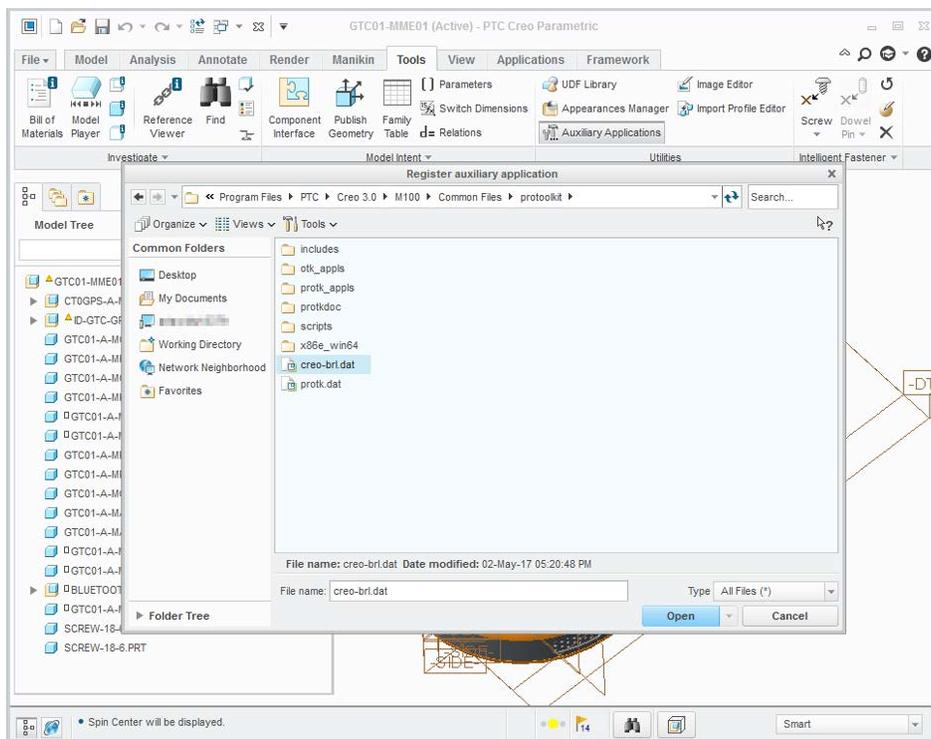


Fig. 10 In the File Selection dialog, navigate to the PTC hierarchy Common Files/toolkit directory. Select the file creo-brl.dat. This file will be in the directory if the plugin was installed correctly.

Approved for public release; distribution is unlimited.

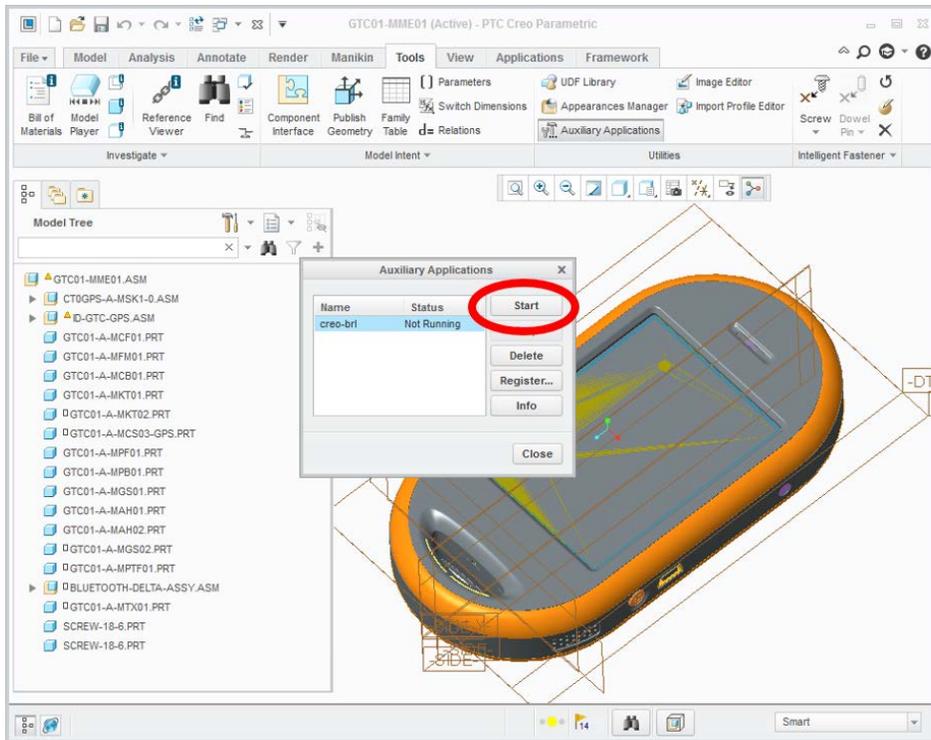


Fig. 11 Once registered, the plugin can now be started

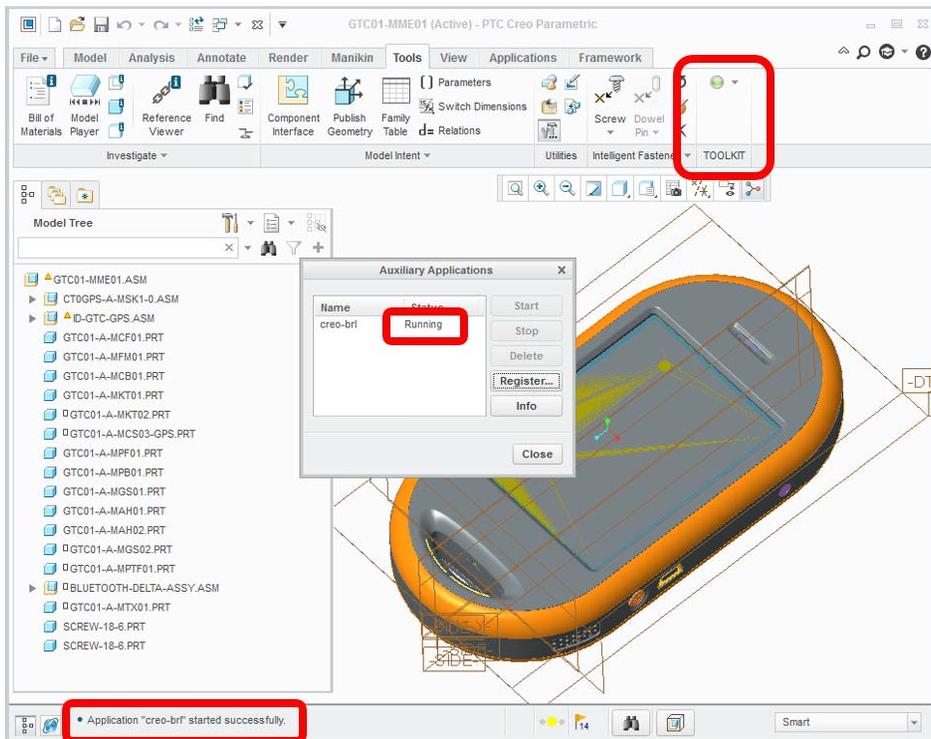


Fig. 12 Screenshot of successful startup: running report from the Auxiliary Applications dialog, a new “TOOLKIT” group in the tool bar, and a message reporting success in the text status window

Approved for public release; distribution is unlimited.

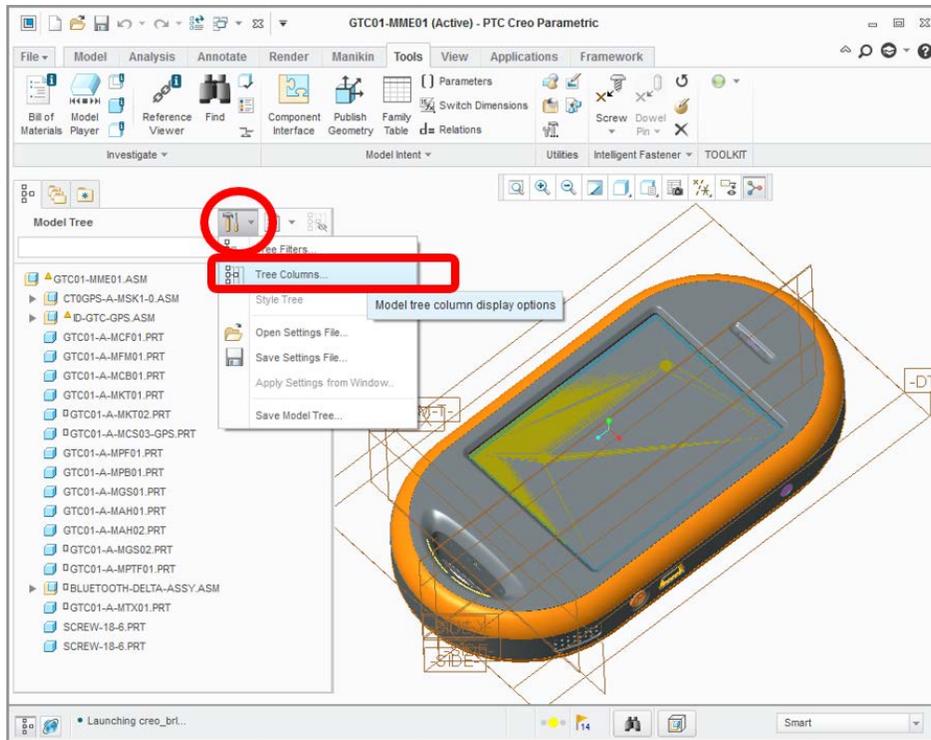


Fig. 13 Select Tree Columns in the Model Tree view to display data in Creo identifying significant metadata on objects for preservation

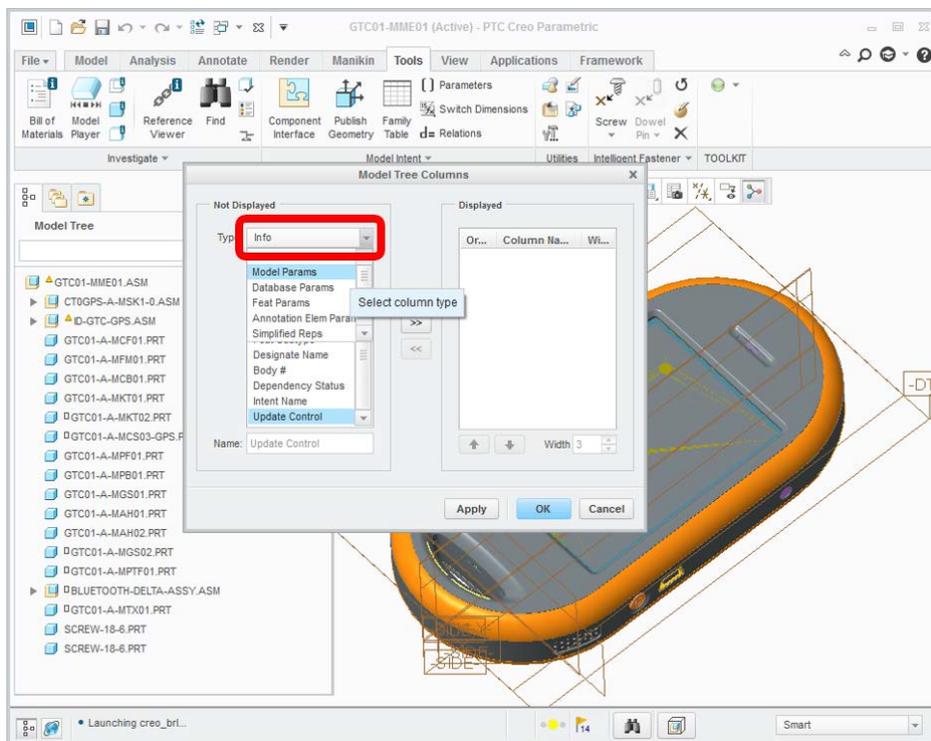


Fig. 14 Various types of parameters are available. As an example, select Model Params for the type of data of interest.

Approved for public release; distribution is unlimited.

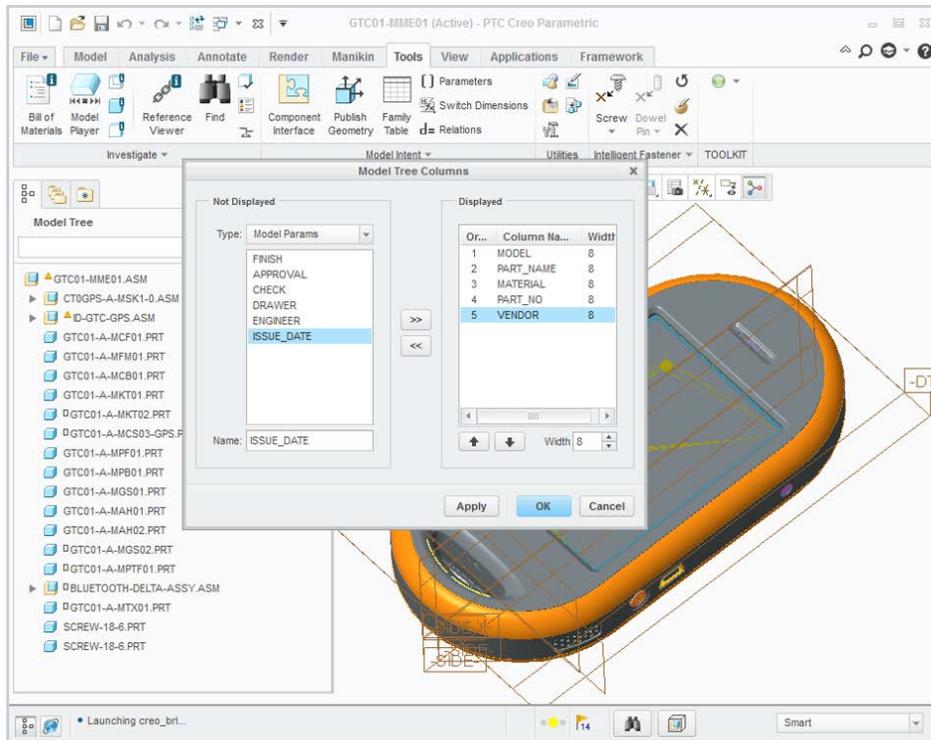


Fig. 15 From the available parameters, select those of possible interest for display in the tree view

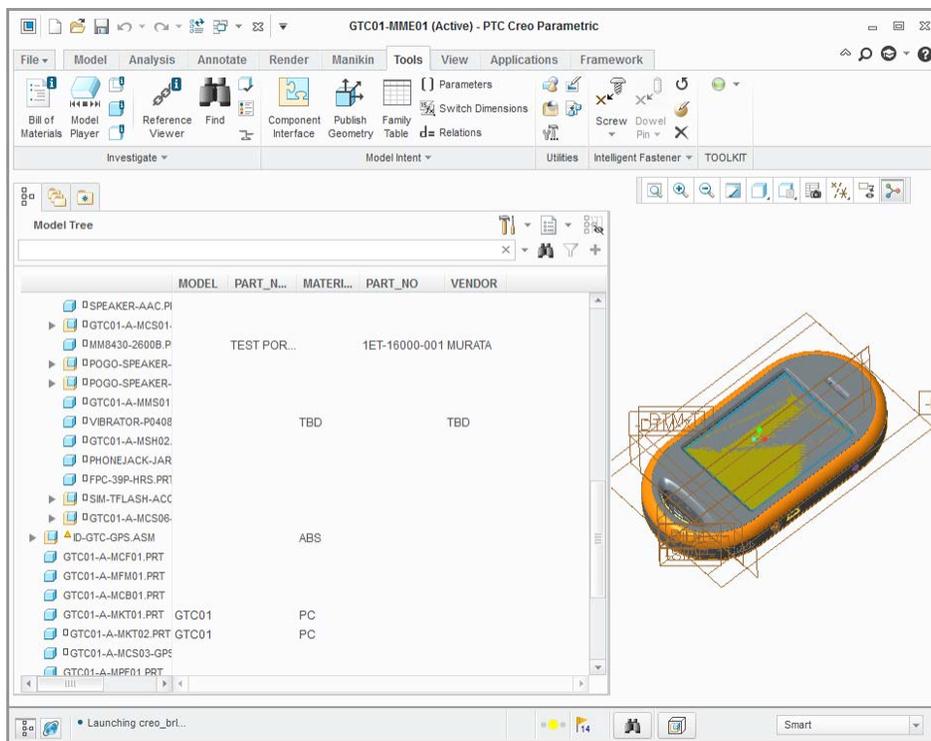


Fig. 16 Inspect the model to identify data are in at least some of the objects

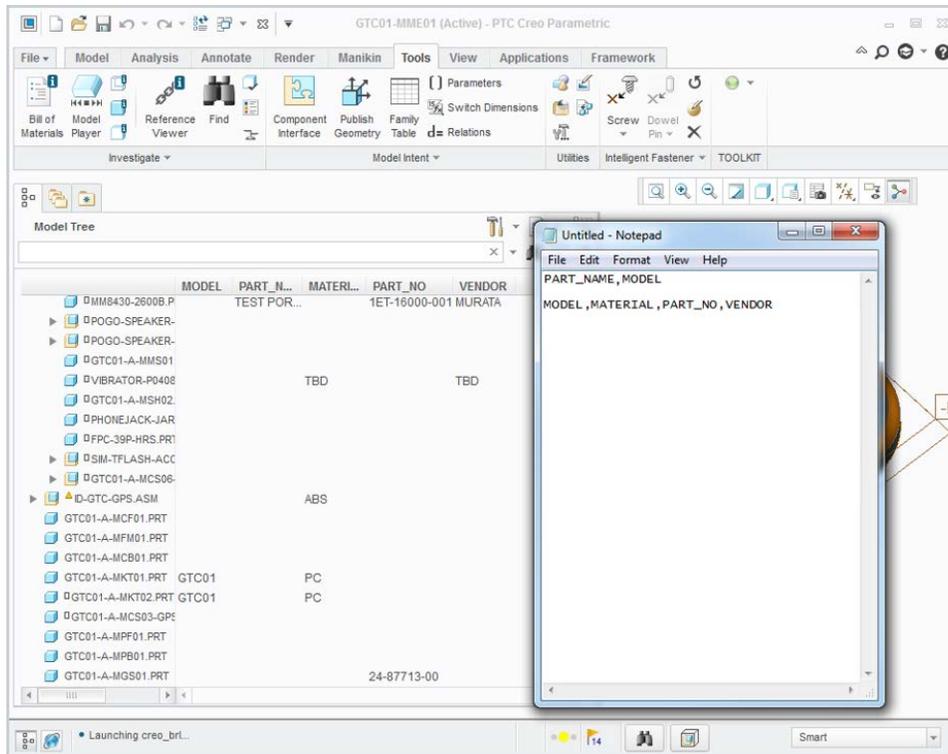


Fig. 17 Construct 2 comma-separated lists of attributes using a text editor. The first indicates the parameters to use when constructing objects names. The second lists parameters to preserve as attributes on the corresponding BRL-CAD objects.

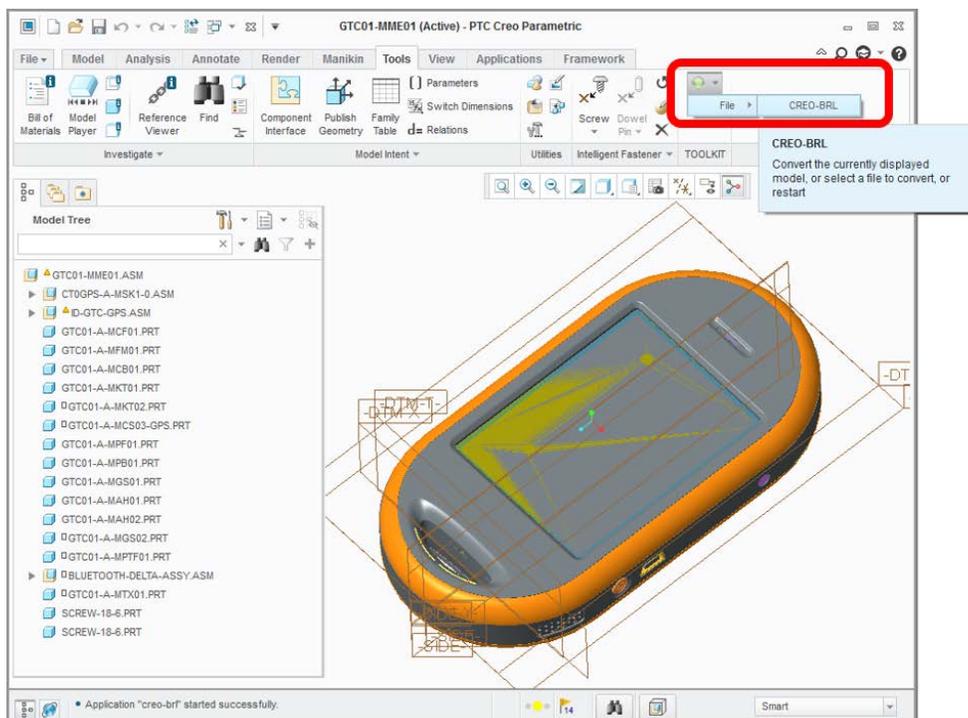


Fig. 18 Select File/Creo-BRL to run the converter

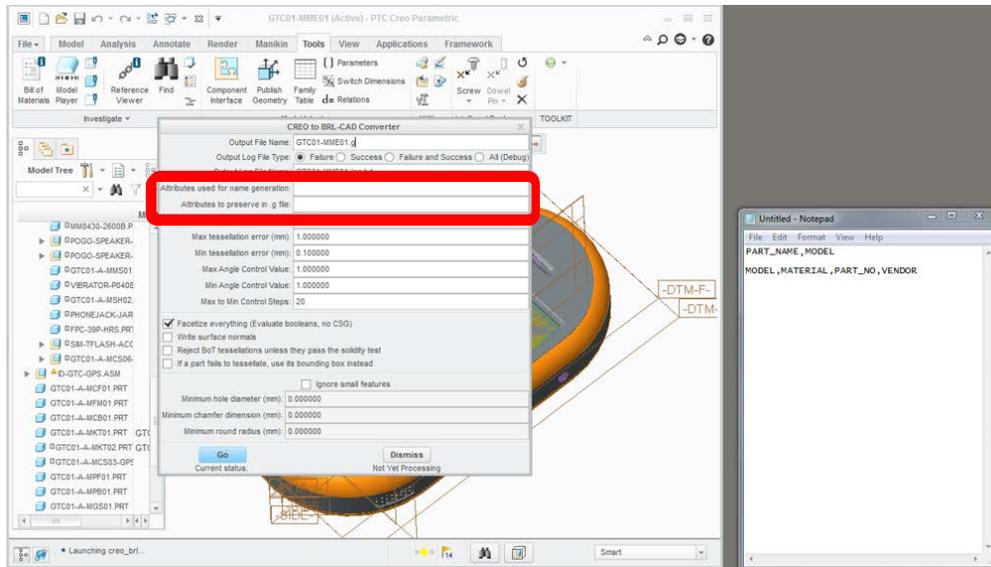


Fig. 19 The converter dialog offers a variety of settings that impact the conversion process. Populate the various data entry boxes in the lists of Creo model parameters. Select All for verbose logging. (This is not necessary, but is sometimes useful when debugging problems.)

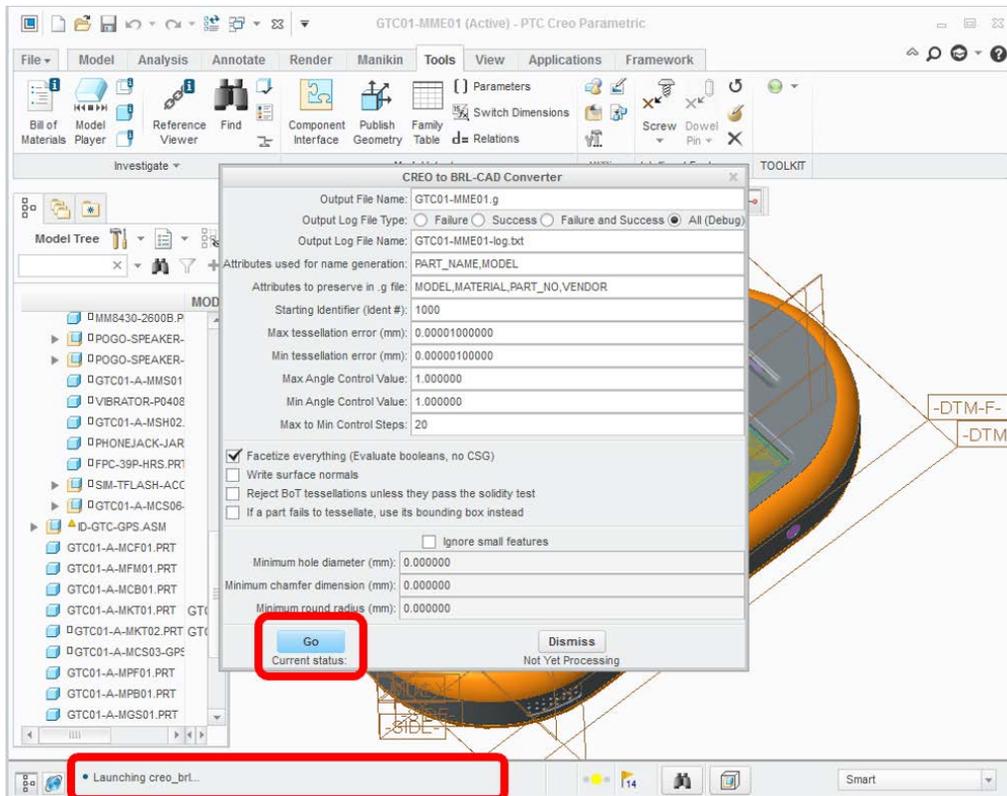


Fig. 20 Once the desired settings are selected, select Go to run the conversion. As the conversion is running, messages should be displayed in the status bar. (This can sometimes fail if the conversion hits an interface update blocking operation, but generally, it should provide a useful sense of how the conversion is progressing.)

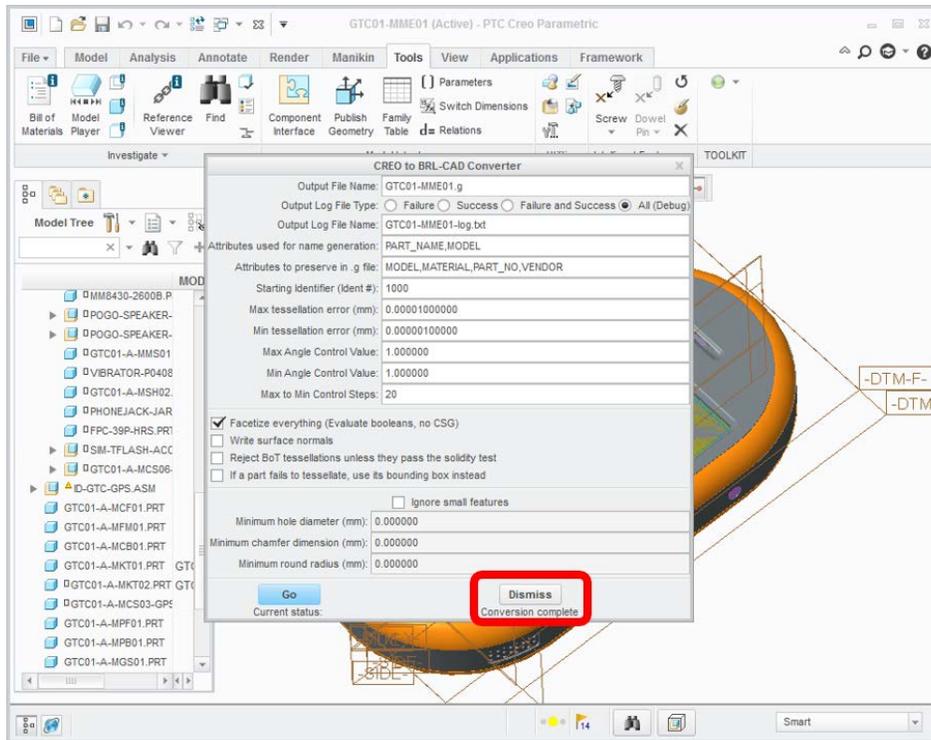


Fig. 21 Once the conversion is complete, select Dismiss

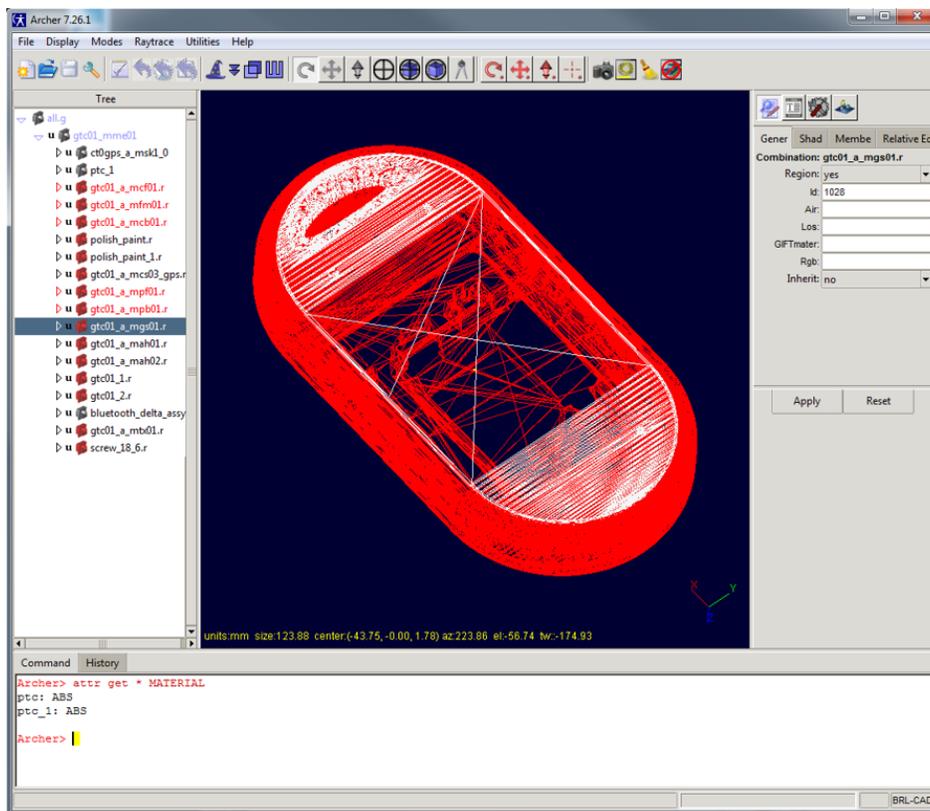


Fig. 22 Inspect the resulting .g file to ensure the result is as expected

Approved for public release; distribution is unlimited.

4. Other Features of the Creo to BRL-CAD Converter

Users of the plugin dialog will note that a number of data entry blocks are not documented in the previous section's conversion. Although these settings will normally work using their default values, the figures and descriptions in this section briefly document the purpose of each option so users will know when they might want to adjust them.

The starting identifier is used to set the initial region ID used when creating regions to hold objects (Fig. 23). The converter begins with this ID number and then increments the ID for each new region created during conversion.

The screenshot shows the 'CREO to BRL-CAD Converter' dialog box. The 'Starting Identifier (Ident #):' field is highlighted with a red box and contains the value '1000'. Other fields include Output File Name (GTC01-MME01.g), Output Log File Type (Failure), Output Log File Name (GTC01-MME01-log.txt), and various tessellation control parameters like Max tessellation error, Min tessellation error, Max Angle Control Value, Min Angle Control Value, and Max to Min Control Steps. There are also checkboxes for 'Facetize everything', 'Write surface normals', 'Reject BoT tessellations', and 'If a part fails to tessellate, use its bounding box instead'. A 'Go' button is at the bottom left, and a 'Dismiss' button is at the bottom right. The current status is 'Not Yet Processing'.

Fig. 23 Starting identifier number

There are 5 settings that all pertain to tessellation, as identified in Fig. 24. These define the range of settings that may be used by the converter when trying to tessellate NURBS geometry, in order from least accurate (and smallest) to most accurate (and largest). Modelers may wish to experiment with these settings on individual parts to get a feel for their behavior.

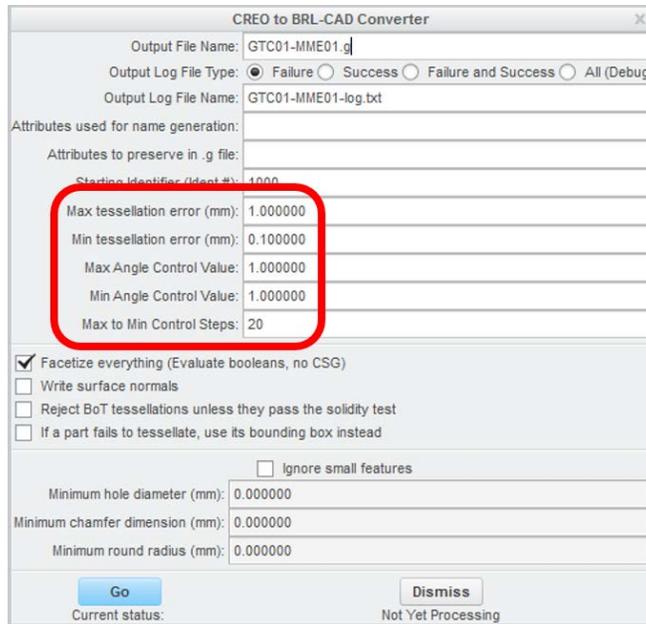


Fig. 24 Tessellation settings

The converter does have some logic defined that will attempt to recognize features defining holes in solids and replace them with constructive solid geometry (CSG) Boolean shapes, but this feature is disabled by default (Fig. 25). It is somewhat experimental, and modelers are cautioned that applications requiring only triangle meshes cannot use outputs generated with this option disabled.

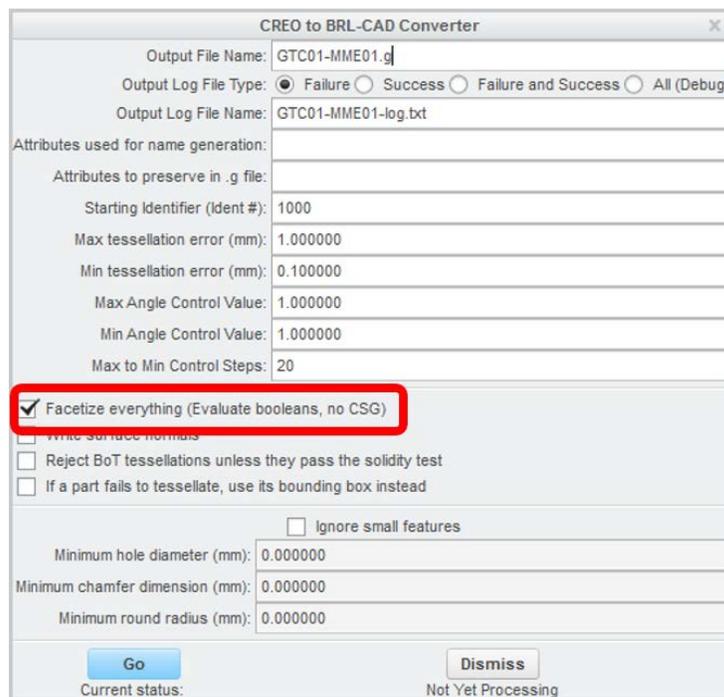


Fig. 25 Disable use of CSG in output

Surface normals on triangles are not written out to bag-of-triangles (BoT) solids by default (Fig. 26), but the BRL-CAD BoT primitive does support them. They will be saved during conversion if this option is enabled.

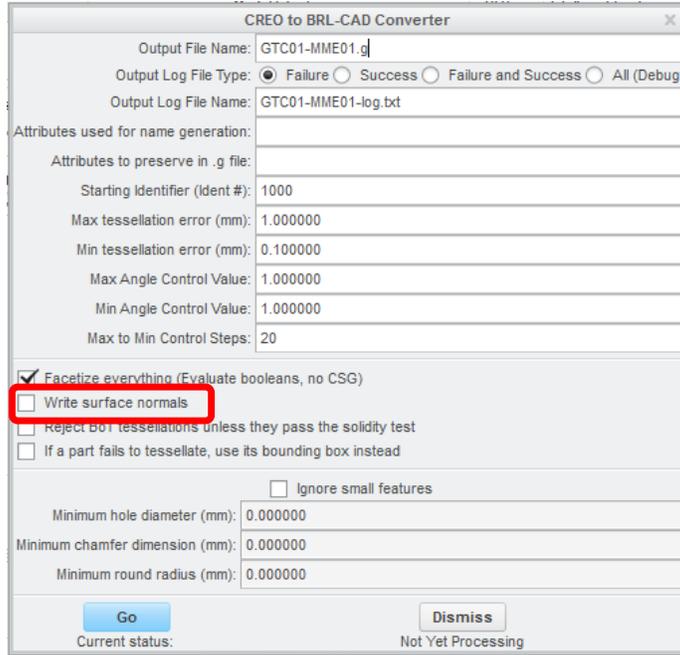


Fig. 26 Write surface normals when outputting triangle meshes

Rejecting nonsolid BoT tessellations (Fig. 27) is an experimental feature, which may be overly aggressive in rejections as currently implemented.

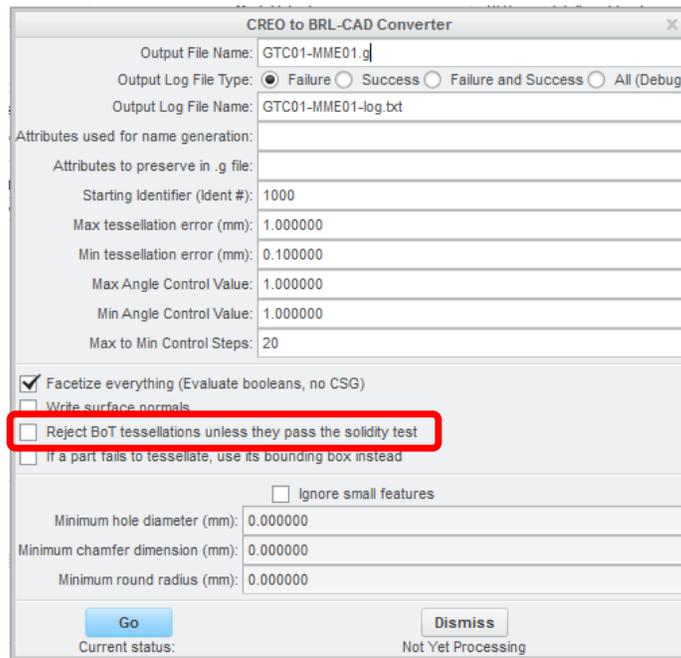


Fig. 27 Reject nonsolid object tessellations

The converter can optionally use the bounding box of an object to create an ARB8 primitive if all tessellation options fail (Fig. 28), but generally, the resulting shape will be too crude of an approximation of the original shape for V/L analysis purposes. This option is primarily for debugging but can also be used to generate a more complete conversion of the geometry hierarchy, with the bounding box primitives indicating objects requiring manual work to complete.

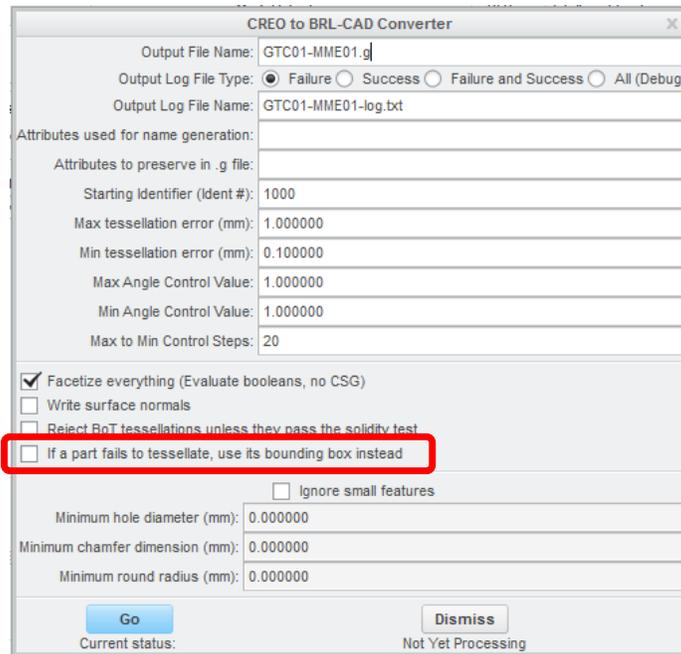


Fig. 28 Use bounding boxes to represent objects if mesh generation fails

Creo Parametric provides the ability to suppress features on solids (Fig. 29), which in effect allows the converter to interact with a simplified approximation of the fully detailed solid. This can allow successful tessellation in cases where the fully featured model will not succeed, and it will tend to generate simpler meshes when the tessellation does not have to use many small triangles to approximate fine details. The check button enables suppression, and once enabled, threshold parameters may be entered for holes, chamfers, and roundings to determine which features end up suppressed. Enabling this process will 1) slow the conversion if many features need to be suppressed and 2) may damage the working copy of the model if features cannot be restored post-conversion. In the latter case, it is important to quit Creo Parametric after conversion is complete without saving the Creo model (and write the inadvertent changes back to the on-disk copy).

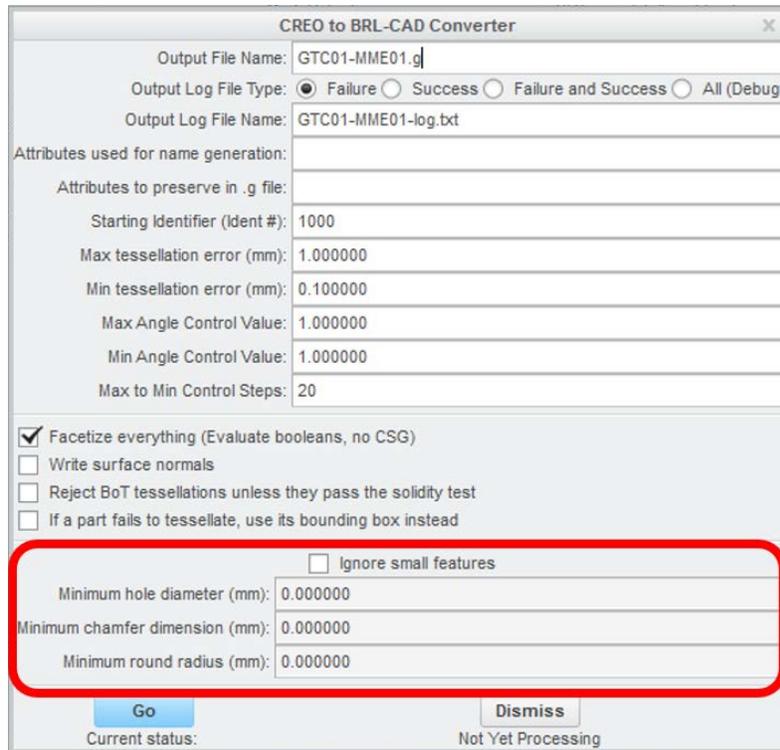


Fig. 29 Creo Parametric provides the ability to suppress features on solids

5. Developers Notes: Working with Creo Parametric

Important points for developers:

- Match your version of Visual Studio to the version used to compile the specific version of Creo Parametric you are targeting—for version 3, Visual Studio 2012 with the v110_xp toolset).
- Ensure you install the development library—by default, it is not enabled for installation in Creo Parametric 3 (See Figs. 30 and 31).
- You will need read-write access to the Creo Parametric directories for development—the plugin cannot be run successfully without performing the installation step, even for debugging purposes.

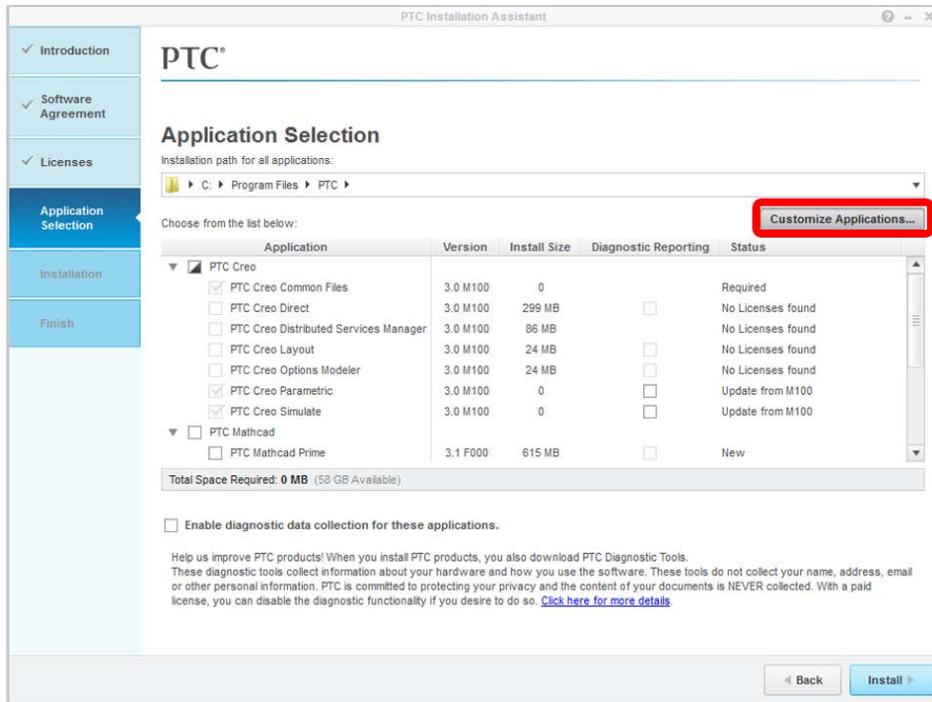


Fig. 30 When you reach the Application Selection stage of Creo Parametric installation, locate the Customize Applications button

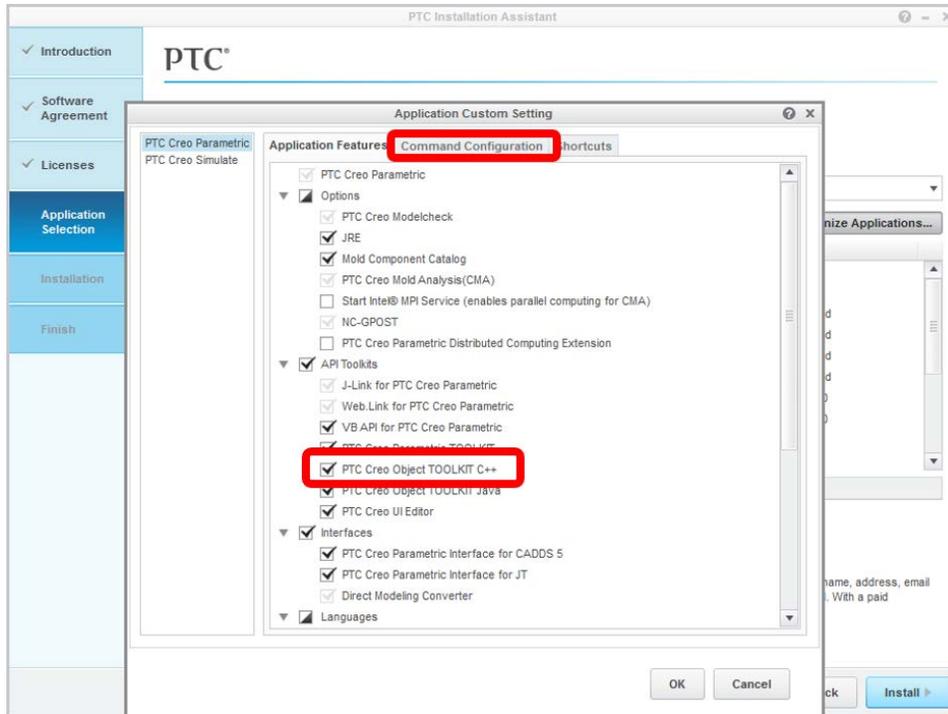


Fig. 31 Under the **Command Configuration** tab, locate the **PTC Creo Parametric TOOLKIT** and ensure it is enabled for installation. If you wish to have **Creo Parametric** load the **TOOLKIT** license at startup (you do if you are doing any major development) update the configuration in the **Command Configuration** tab. See the **Creo** documentation for more details.

Once **Creo Parametric** is correctly set up, it is time to compile the plugin. The source code for this plugin is part of a standard **BRL-CAD** source archive, located in the directory `src/external/Creo`. Unlike most of **BRL-CAD**'s tools, the **Creo** plugin is designed to be built as a stand-alone project. The **Creo** plugin uses the **CMake** build tool to manage its configuration process and **Microsoft Visual C++** to perform the actual compilation, as illustrated in Figs. 32–38.

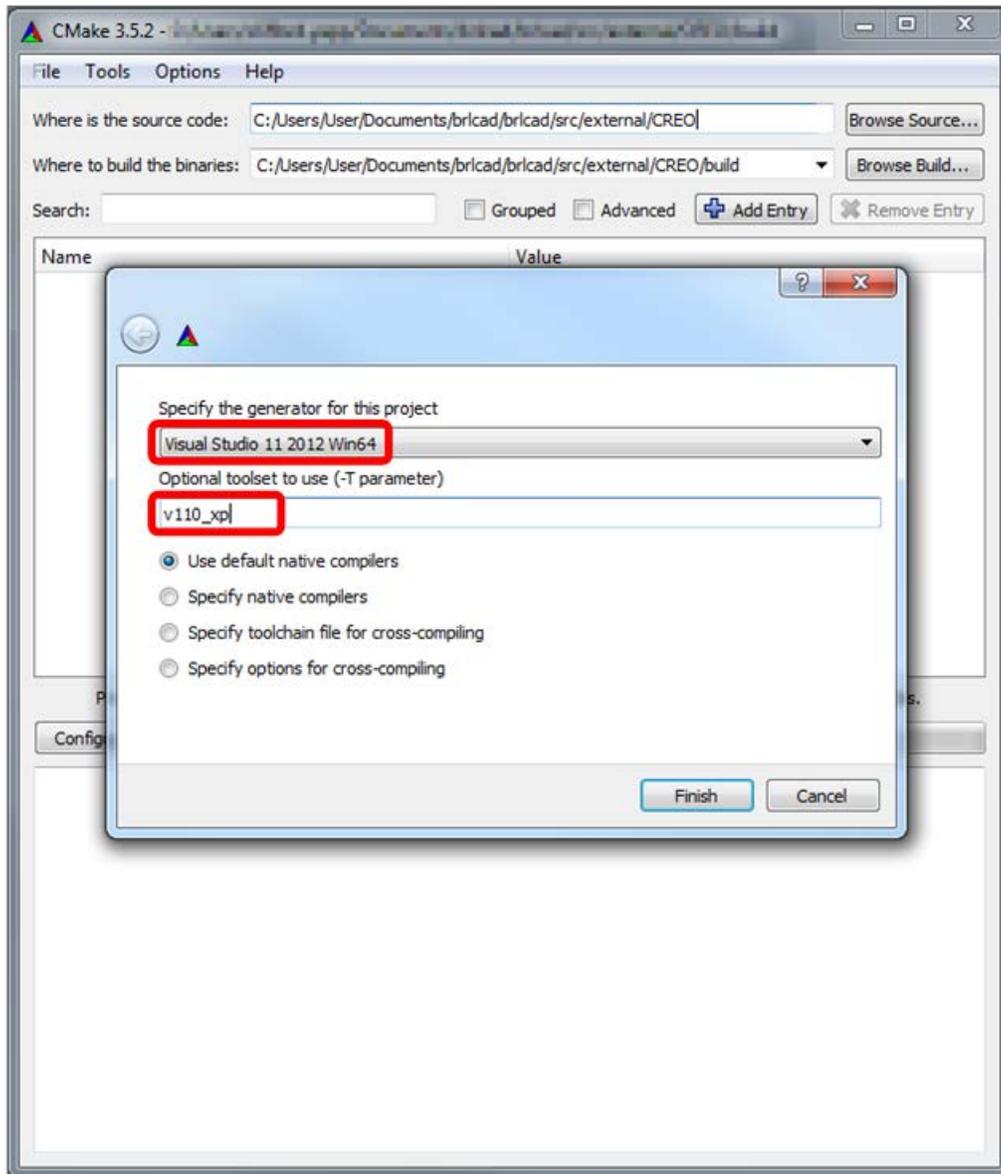


Fig. 32 Visual Studio version and toolset are both set in the initial CMake screen

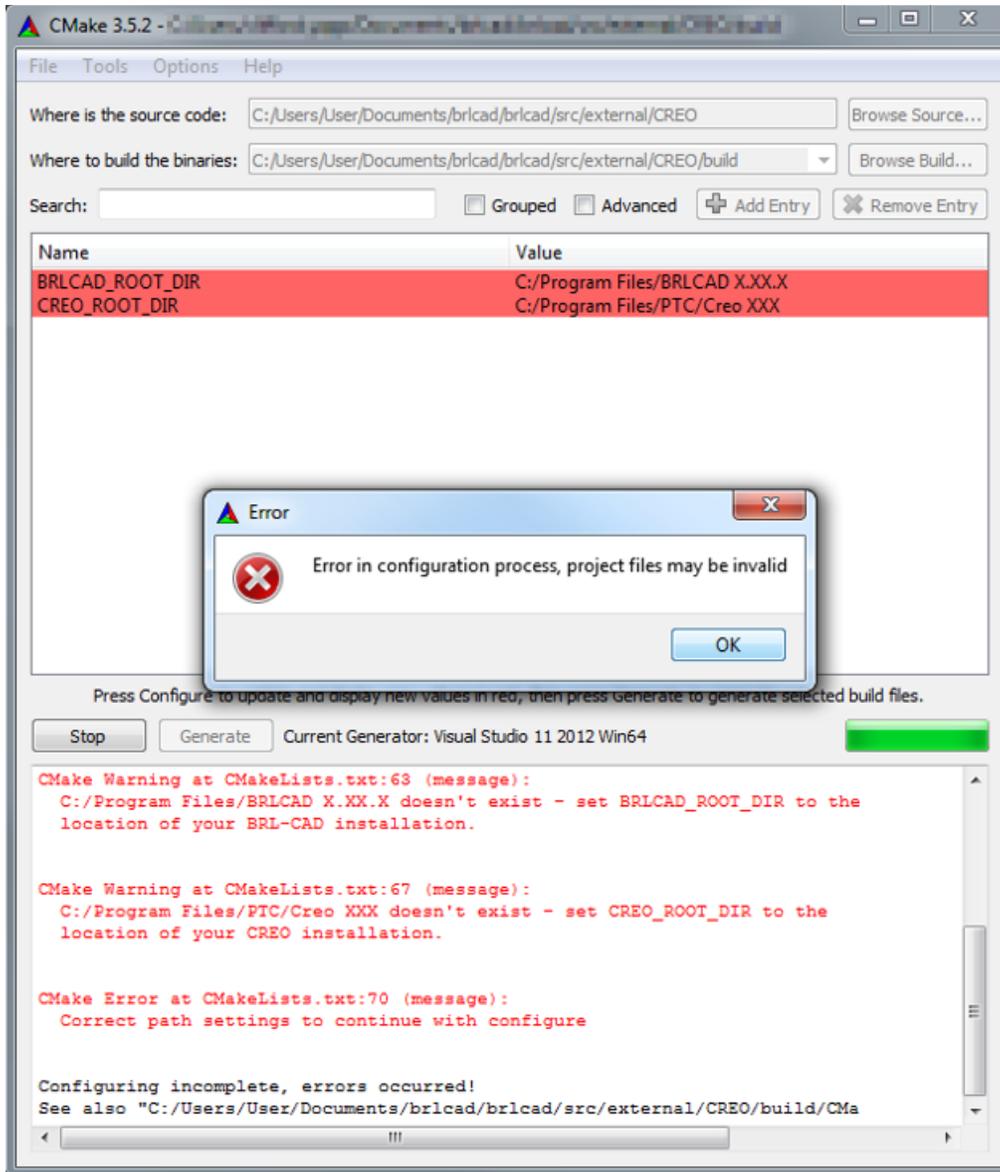


Fig. 33 The first time Configure is run, the process will halt with an error. This is expected behavior, because the locations of BRL-CAD and Creo Parametric have not yet been specified. (Both are required for a complete configuration process to succeed.)

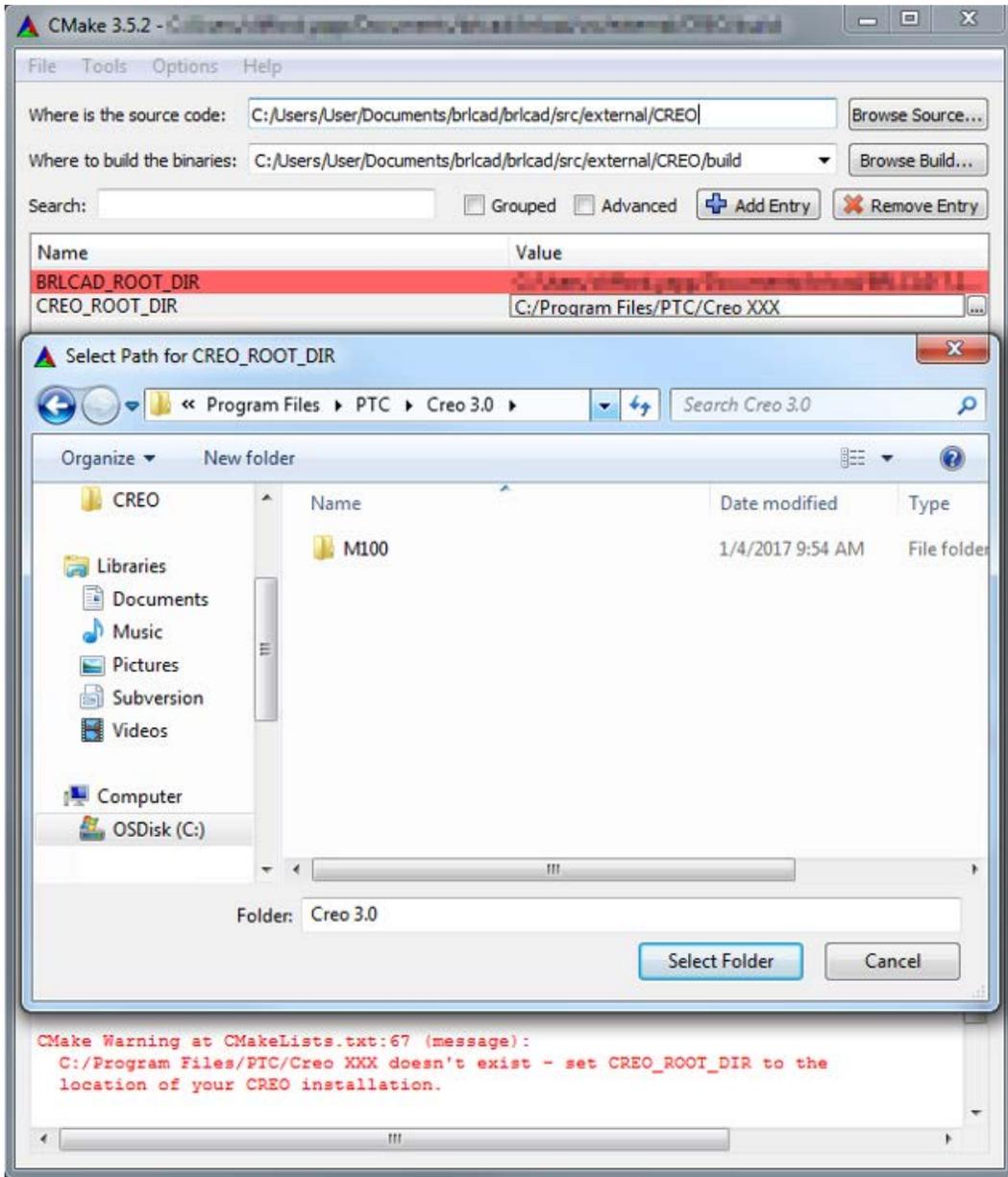


Fig. 34 For BRLCAD_ROOT_DIR and CREO_ROOT_DIR, select the correct directories to identify the versions of Creo Parametric and BRL-CAD being used to build the plugin

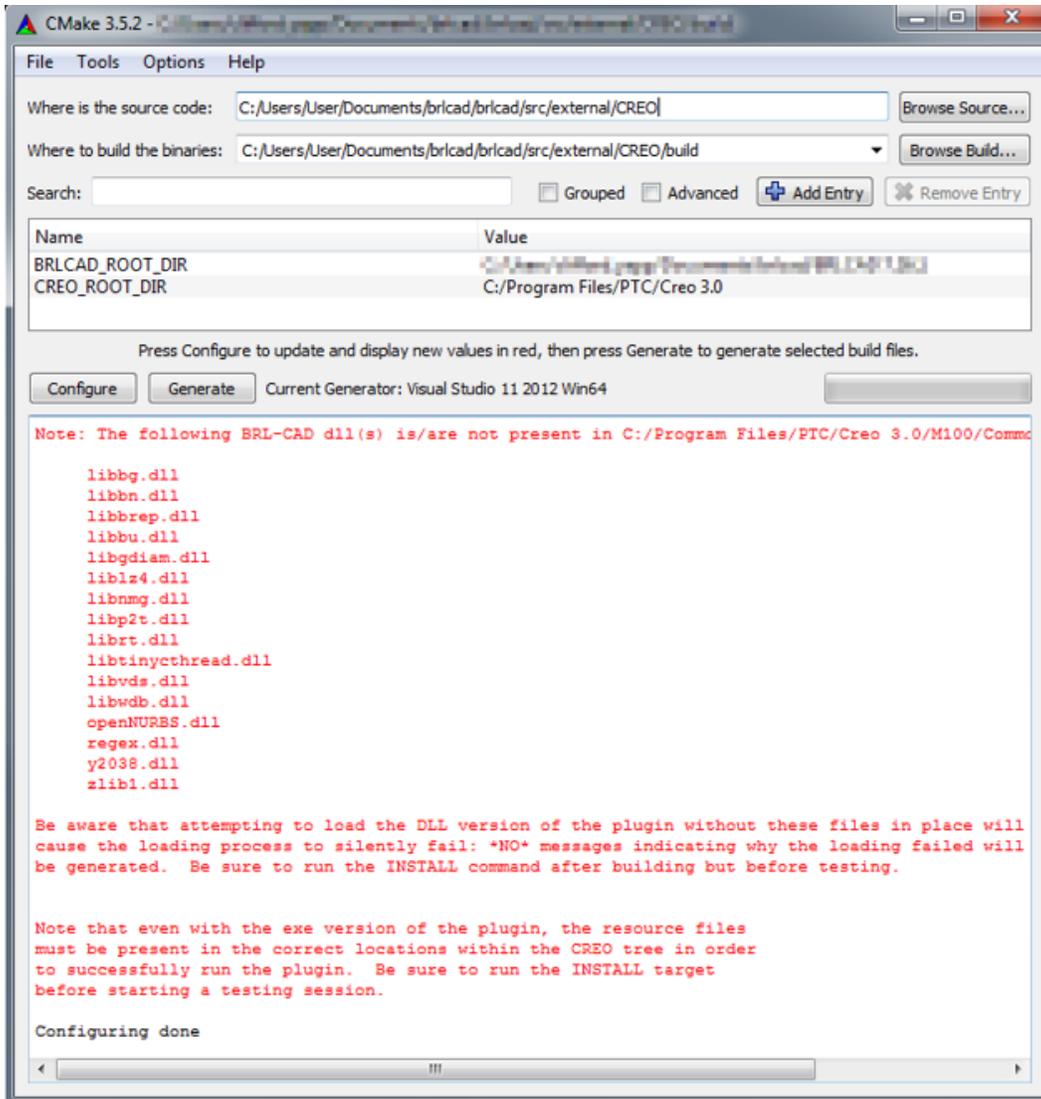


Fig. 35 Run Configure again, and the process should complete. Select Generate to produce a Visual Studio project. Be sure you have read and understood the messages generated during the configuration process—they are intended to alert you to situations that may result in a failure of the converter plugin to successfully run, or otherwise compromise effective software development work on the converter.

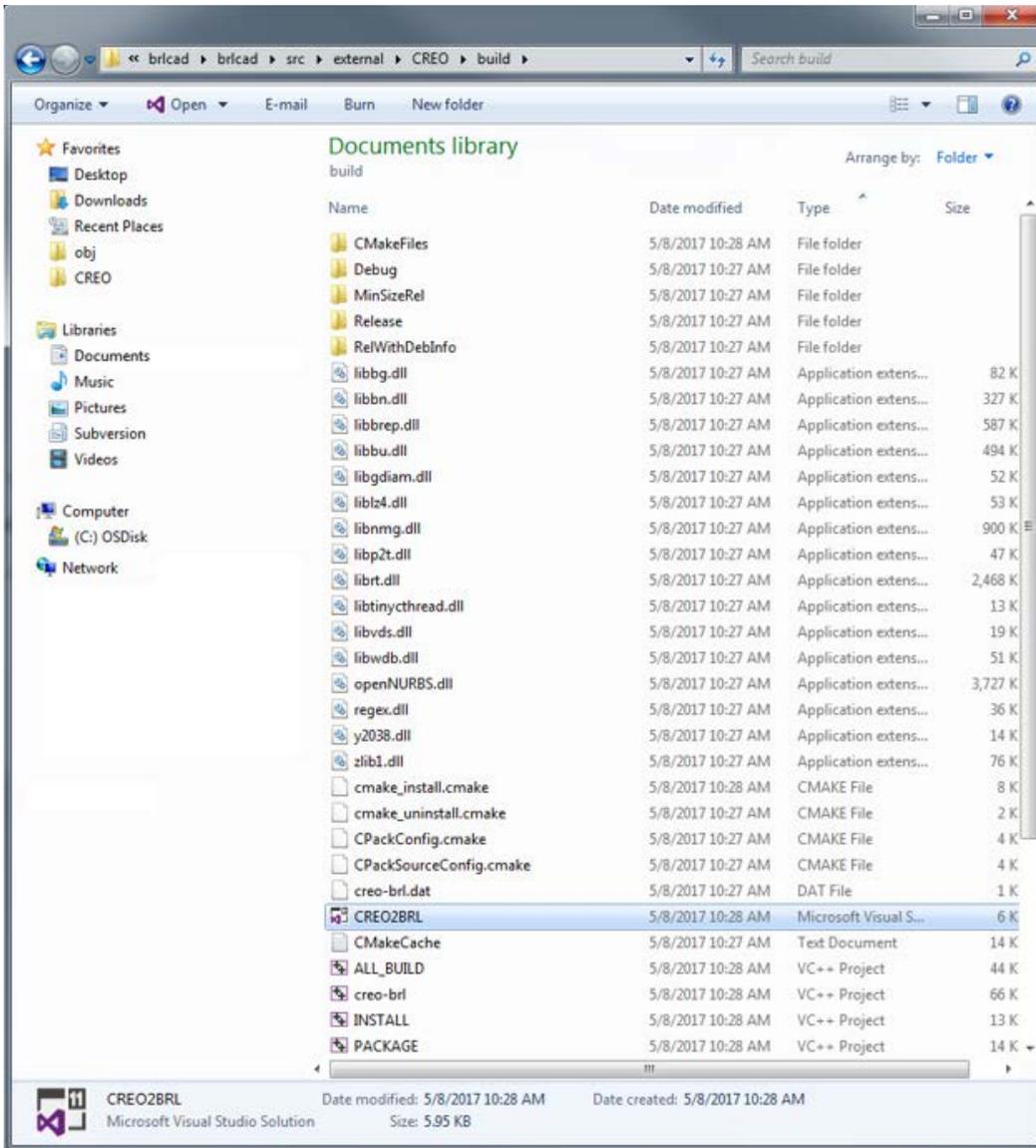


Fig. 36 Select the generated file “CREO2BRL” to open the Visual Studio project

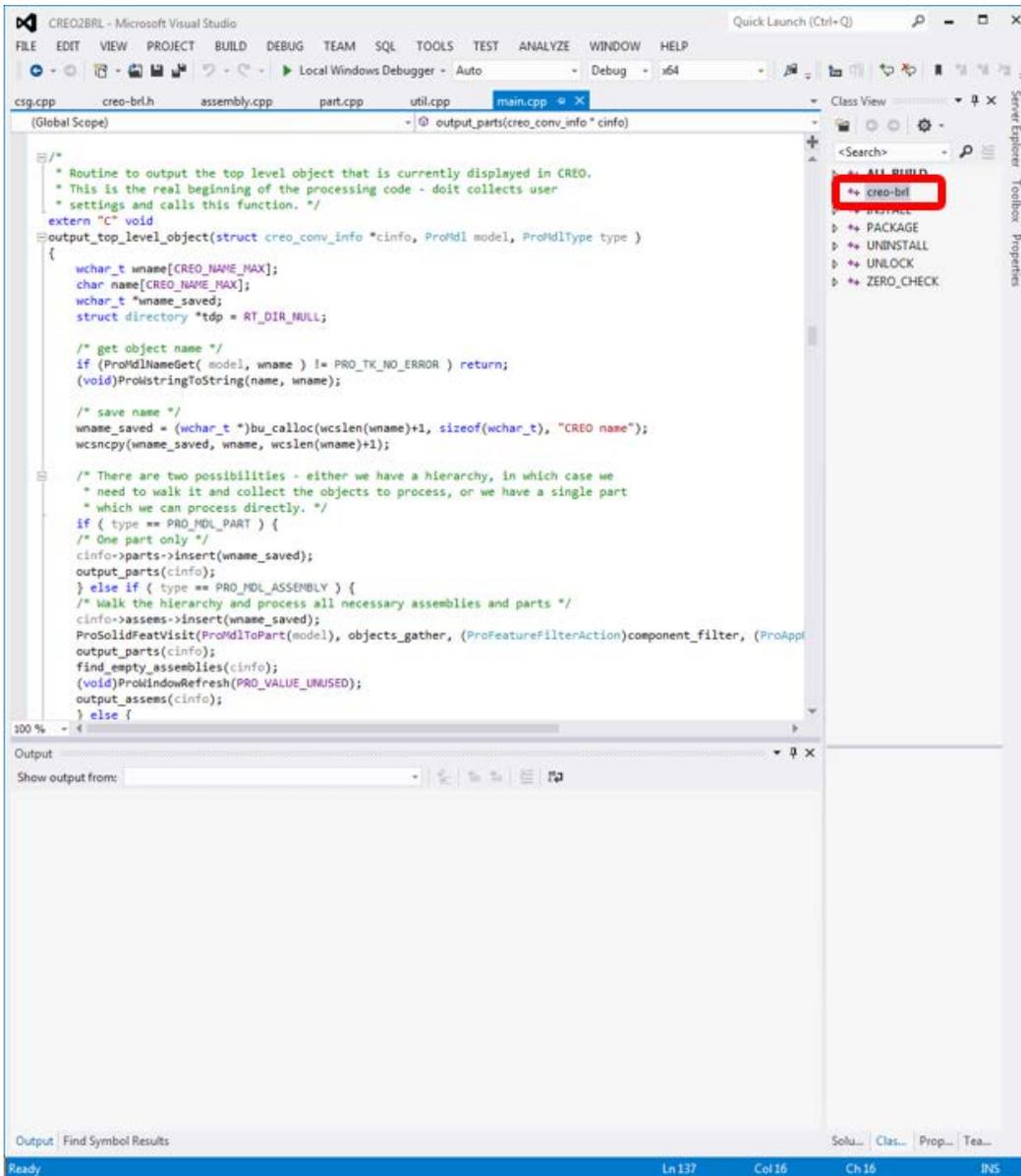


Fig. 37 The target list should include `creo-brl`, which is the primary target to build. Although it is not required for building, opening the converter source code files in Visual Studio is a good preparation step if you expect to set breakpoints or step through the sources during debugging.

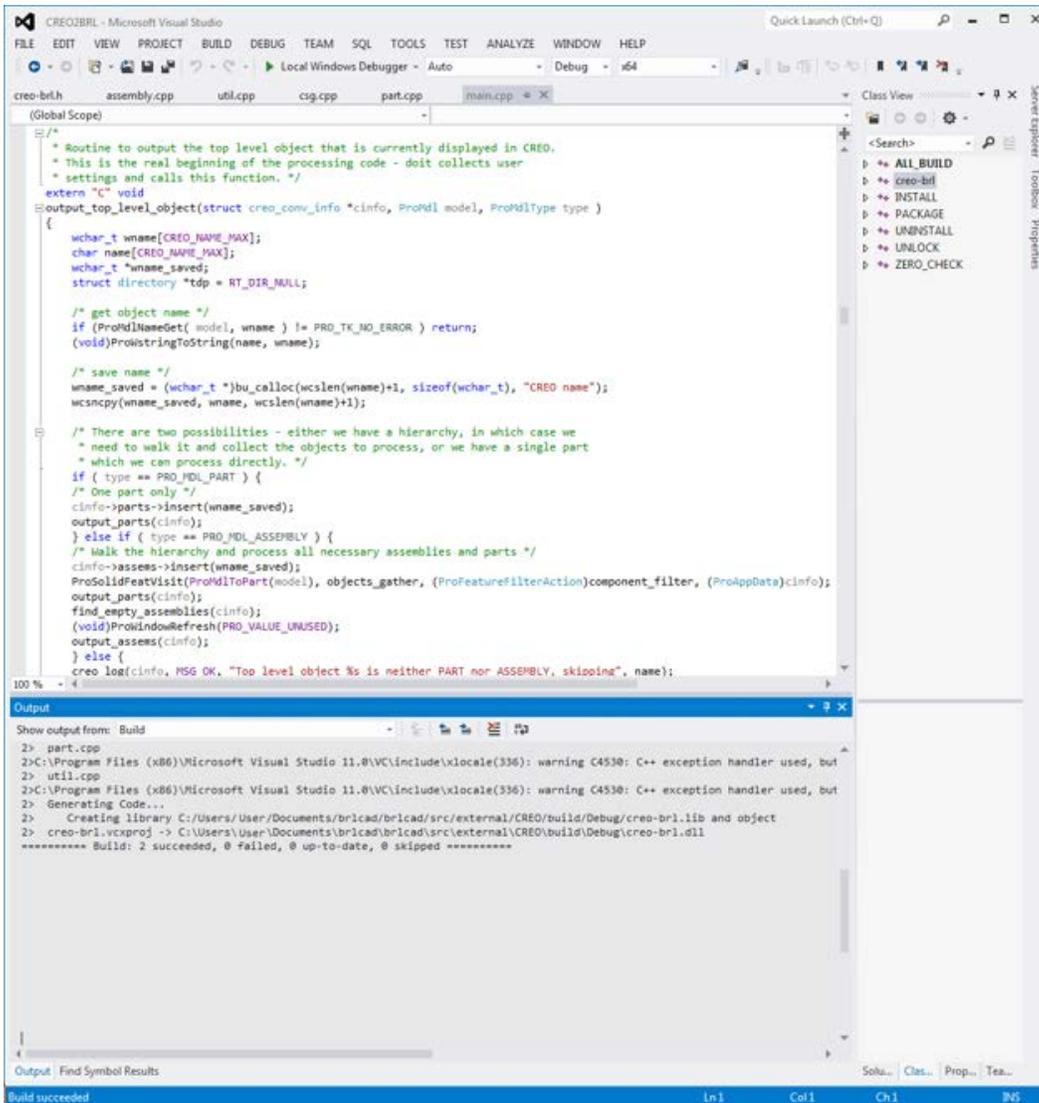


Fig. 38 Building the target should succeed and produce `creo-brl.dll`. If not, debugging work starts now.

After producing `creo-brl.dll` successfully, the next step depends on what licensing the developer's Creo Parametric is set up to use. The CMake process issues a warning if the user's Creo Parametric settings are not configured to load the Pro/TOOLKIT license at Creo startup. If Creo Parametric does not load the necessary toolkit, the plugin will eventually fail to load, notifying the user of a locked DLL. If this is not what is expected, correct the Creo Parametric installation to load Pro/TOOLKIT at startup. If you prefer not to do so (i.e., if you only expect to compile and test once) run the UNLOCK target now, before running the INSTALL target. This is not recommended when doing any sort of extensive development, as it will lock a Pro/TOOLKIT license for 15 min—for development, it is much better to load Pro/TOOLKIT in Creo Parametric at run time.

For the standard situation where a developer needs to debug a running plugin, Figs. 39–41 illustrate how to use a compiled plugin and Visual C++ to step through the source code.

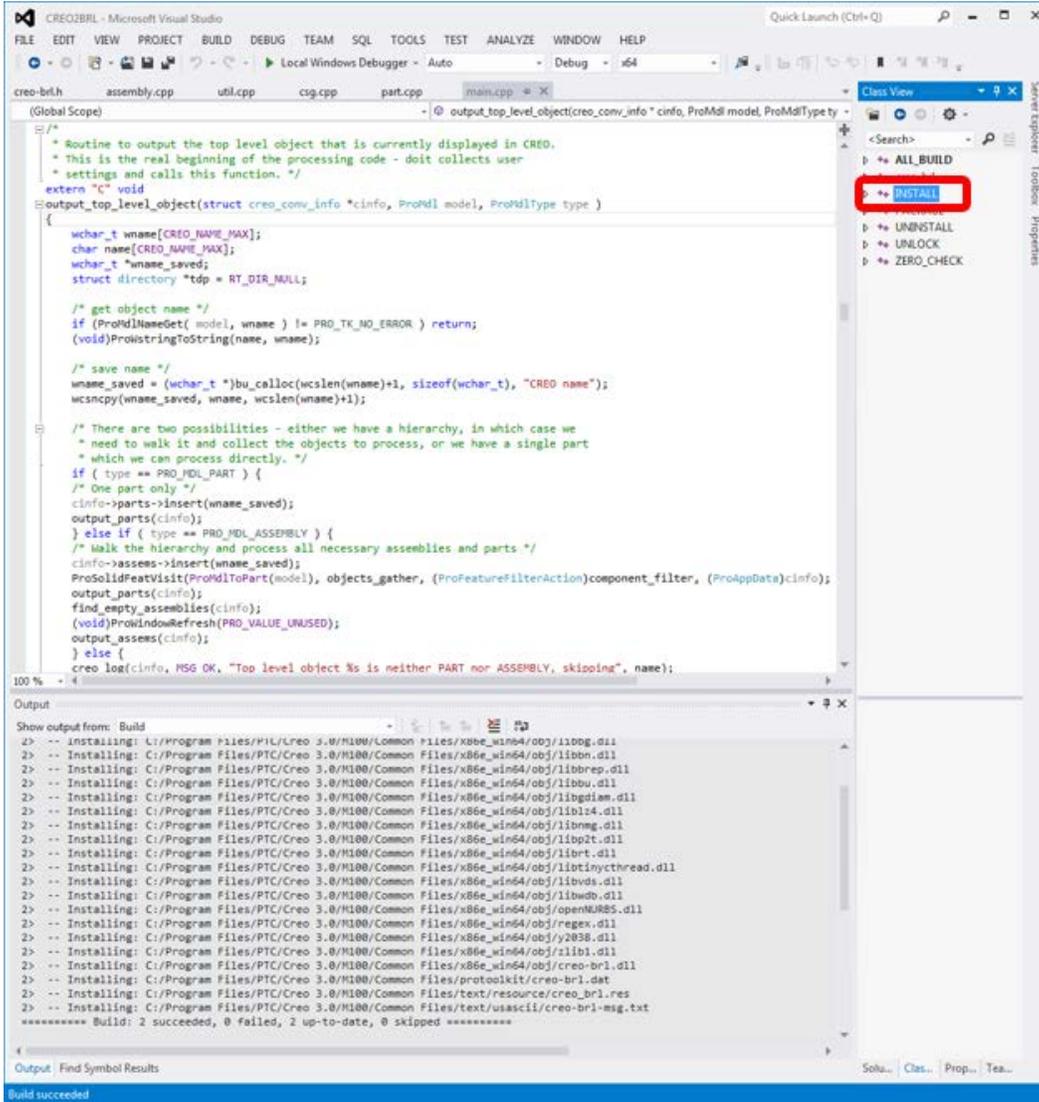


Fig. 39 Before the plugin can be loaded, even for testing, it is necessary to run the INSTALL target to make sure all of the required files are in their correct locations in the Creo Parametric directories

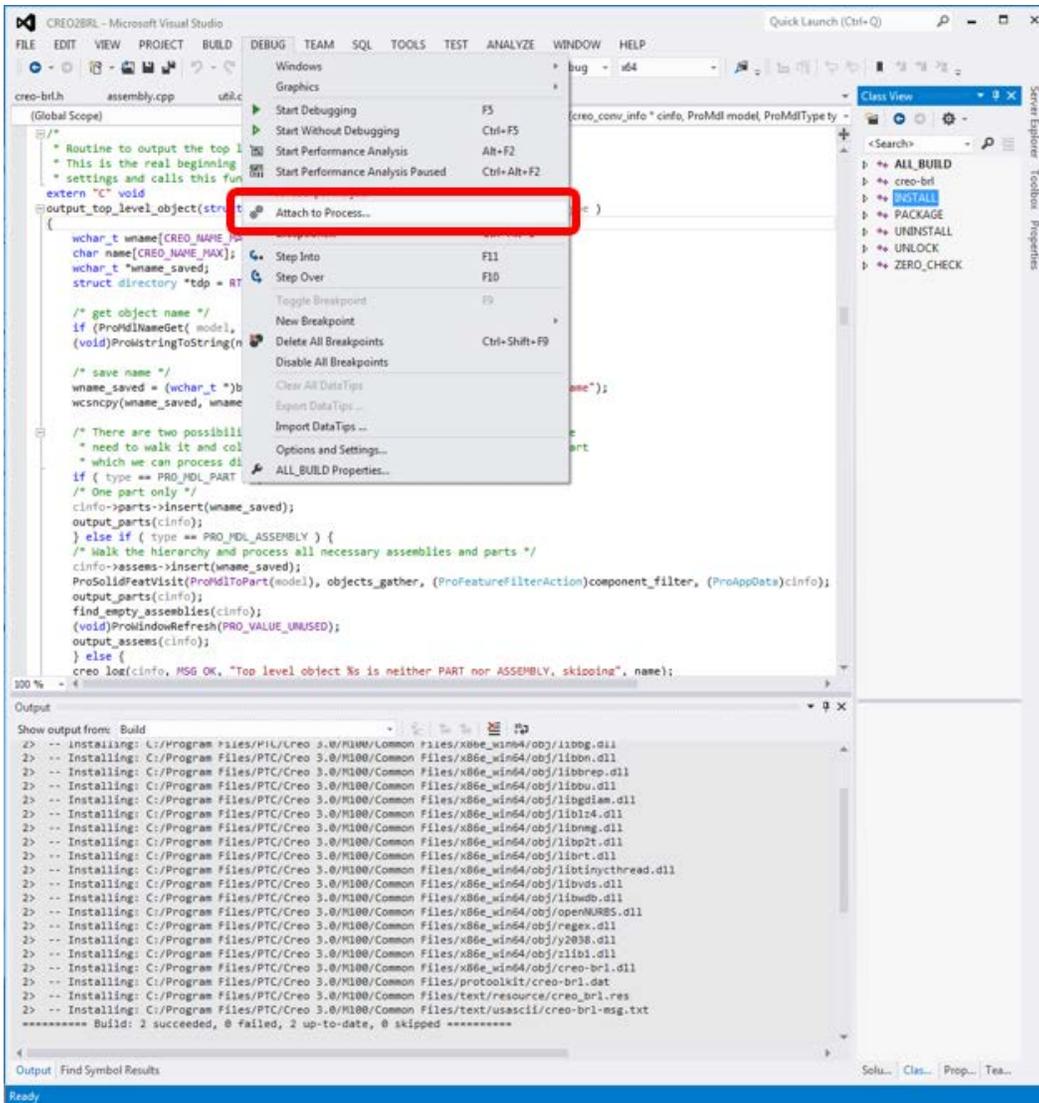


Fig. 40 If Creo Parametric and the plugin are now running and ready for debugging, then attach Visual Studio's debugger to the running process

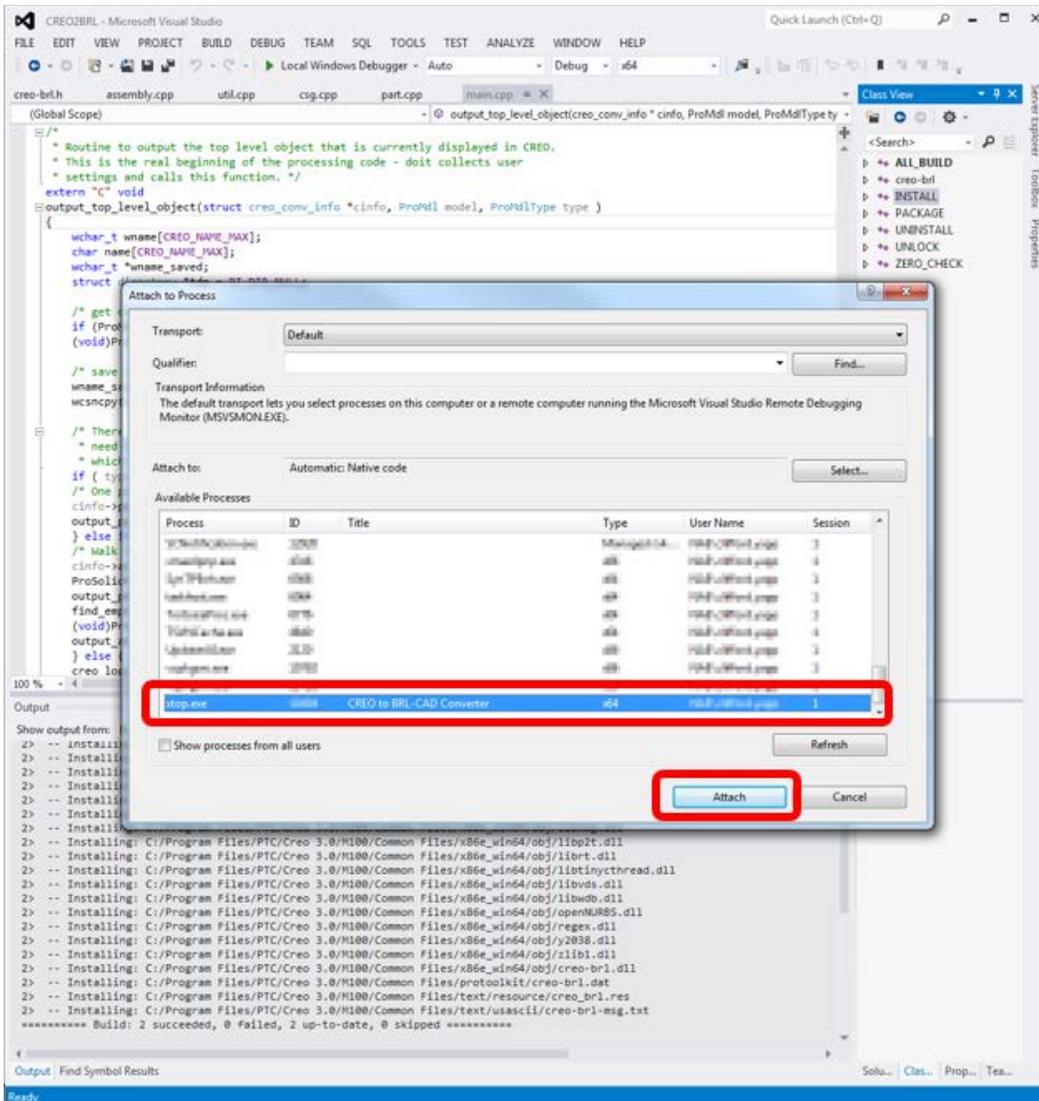


Fig. 41 From the list of processes, select the one that shows either Creo Parametric (if the plugin is not yet running) or Creo to BRL-CAD (if the plugin is running) and select Attach. Once this is complete, the running plugin will respect break points defined in Visual Studio and the conversion process can be debugged.

Once the plugin is running to a developer’s satisfaction, the final step is to prepare a redistributable version of the plugin that users can install. The CPack tool is used to create archives, but it is crucial for developers to remember to perform the “UNLOCK” step before creating the archives, as illustrated in Figs. 42 and 43.

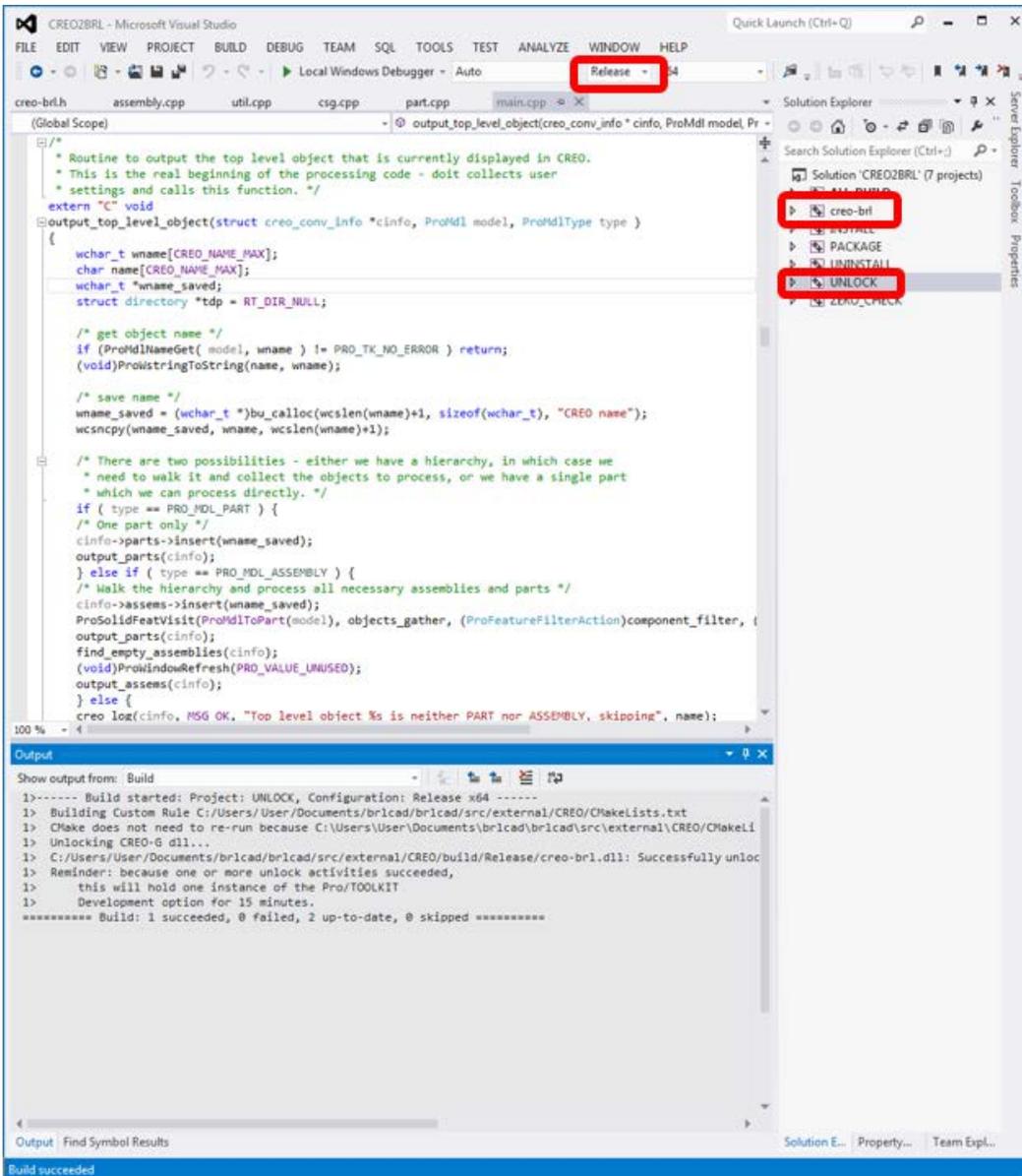


Fig. 42 Once the plugin is deemed ready for distribution, make sure the build type is set to Release, build the `creo-brl` target to make sure everything is up to date, and then run the UNLOCK target. This will lock up a Pro/TOOLKIT license for 15 min, but also results in a DLL that can be used by other Creo Parametric installations.

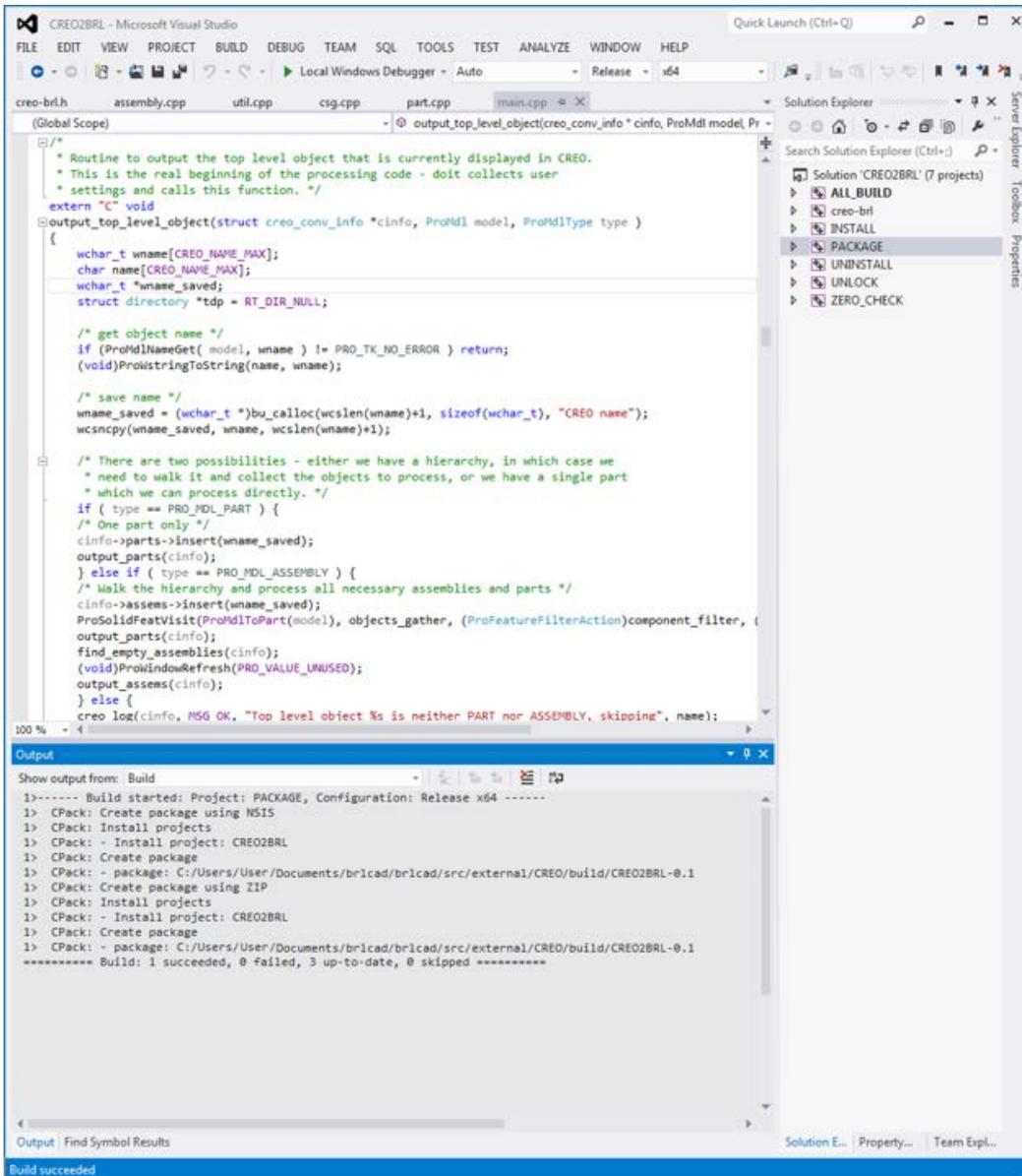


Fig. 43 Once the DLL is unlocked, the PACKAGE target may be used to generate a .zip file and a Nullsoft Scriptable Install System (NSIS) installation executable for distribution

6. Future Work

While the essential functionality of the original Pro/E conversion plugin has been reproduced in Creo Parametric 3 and a number of improvements added, additional work would substantially improve the existing capabilities. Immediately apparent improvements would be to complete the solidity testing routine for mesh generation and add an ability to translate the NURBS boundary representation (B-Rep) solid descriptions over directly to BRL-CAD without tessellation. Longer-term

directions to explore include using BRL-CAD's own tessellation routines as an alternative to those in Creo Parametric for more robust solid generation and applying CSG conversion research⁸ to imported Creo NURBS B-Reps as a means of size reduction.

7. References

1. Guber W, Nagel R, Goldstein R, Mittelman PS, Kalos MH. A geometric description technique suitable for computer analysis of both the nuclear and conventional vulnerability of armored military vehicles. Army Ballistic Research Laboratory (US); 1967 Aug. Report No.: MAGI-6701. Also available at <http://www.dtic.mil/get-tr-doc/pdf?AD=AD0847576>.
2. Hanes PJ, Murray KR, Gwyn DA, Polak HR. An overview and status report of MUVES (modular Unix-based vulnerability estimation suite). Aberdeen Proving Ground (MD): Ballistic Research Laboratory (US); 1988 Jul. Report No.: BRL-MR-3679. Also available at <http://www.dtic.mil/get-tr-doc/pdf?AD=ADA199985>.
3. Anderson J. Pro/E converter. Aberdeen Proving Ground (MD): Army Research Laboratory (US); 2001–2003.
4. Parker R. Pro/E converter update. Aberdeen Proving Ground (MD): Army Research Laboratory (US); 2008–2009.
5. Creo Product mapping. Needham (MA):PTC; c2017 [accessed 2017 May 9]. <http://www.ptc.com/cad/pro-engineer/creo-product-mapping>.
6. Puffer F. Customizing and automating PTC Creo. Munich (Germany): Software Factory; 2014 [accessed 2017 May 9]. http://www.ptcuser.nl/conf_14/presentations/1-2-Customizing and Automating PTC Creo/APIs-2014.pdf.
7. Openmokowiki. CAD models–Openmoko. Openmoko development community. [place unknown]: Wikipedia; 2007–2010 [accessed 2017 May 9]. http://wiki.openmoko.org/wiki/CAD_models.
8. Yapp C. An investigation into conversion from non-uniform rational B-spline boundary representation geometry to constructive solid geometry. Aberdeen Proving Ground (MD): Army Research Laboratory (US); 2015 Dec. Report No.: ARL-SR-0347. Also available at <http://www.arl.army.mil/arlreports/2015/ARL-SR-0347.pdf>.

List of Symbols, Abbreviations, and Acronyms

3-D	3-dimensional
ARL	US Army Research Laboratory
B-Rep	boundary representation
BoT	bag-of-triangles
BRL	Ballistic Research Laboratory
CAD	computer-aided design
CSG	constructive solid geometry
MUVES	Modular Unix-based Vulnerability Estimation Suite
NSIS	Nullsoft Scriptable Install System
NURBS	Non-Uniform Rational B-Spline
Pro/E	Pro/ENGINEER
STL	STereoLithography
V/L	vulnerability/lethality

1 DEFENSE TECHNICAL
(PDF) INFORMATION CTR
DTIC OCA

2 DIRECTOR
(PDF) US ARMY RSRCH LAB
RDRL CIO L
IMAL HRA MAIL & RECORDS
MGMT

1 GOVT PRINTG OFC
(PDF) A MALHOTRA

1 DIR US ARMY EVALUATION CTR HQ
(PDF) TEAE SV
P A THOMPSON

30 DIR USARL
(PDF) RDRL SL
P BAKER
P DISALVO
D BAYLOR
N EBERIUS
T STADTERMAN
RDRL SLB
B BOWEN
RDRL SLB D
R GROTE
RDRL SLE
R FLORES
RDRL SLB E
M MAHAFFEY
RDRL SLB G
N ELDREDGE
M ROTHWELL
J ABELL
J BAKER
V CERICOLE
D KREGEL
J LALONE
E MURRAY
F REYNOLDS
K RICHMAN
M SANDERSON
C STEWART
J THOMPSON
RDRL SLB S
M PERRY
G SAUERBORN
W BOWMAN
R PARKER
C MORRISON
N REED
C YAPP
RDRL SLB W
S SNEAD

INTENTIONALLY LEFT BLANK.