TETRA 4D

Tetra4D Automate

Version 2024

User Guide

Document version: V4.0

Table of Contents

Installation

System Requirements

Prior to any installation, please check with your Tetra4D software provider the requirements and recommendations to run the Tetra4D "Automate".

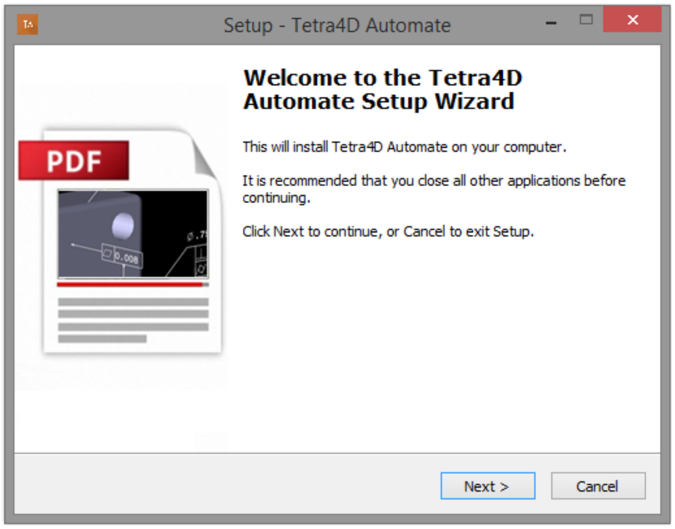
Note that Tetra4D Automate requires templates created with Tetra4D enrich to operate.

Windows

- 1.5GHz or faster processor
- 1GB of RAM
- 350MB of available hard-disk space
- 1024x768 (minimum) screen resolution
- Video hardware acceleration (optional)
- One of the following Microsoft® Windows® operating systems:
  - Windows Server 2008 R2 (32 bit and 64 bit)
  - Windows 7 (32 bit and 64 bit)
  - Windows 8.1 (32 bit and 64 bit)
  - Windows 10 (32 bit and 64 bit)

Installation of Tetra4D Automate

1. Download and run the "Tetra4D Automate" installer. The installer should have been emailed to you along with your license/serial number. You can also find it in your Customer Portal



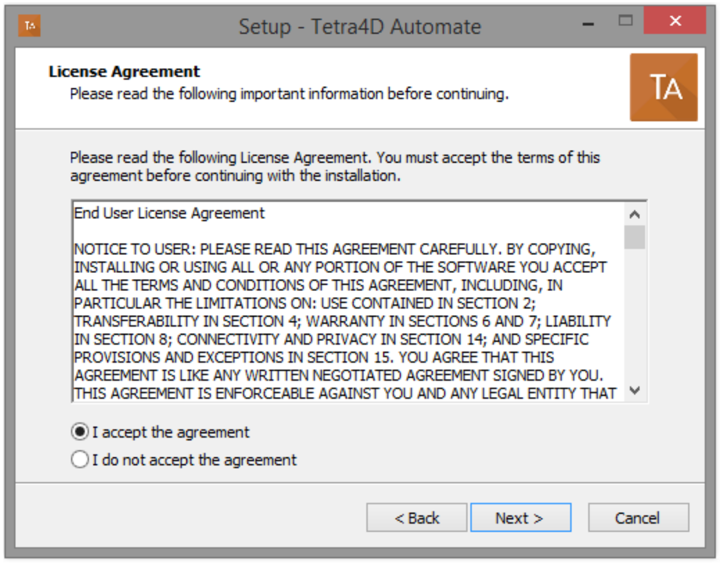
Welcome to the Tetra4D Automate Setup Wizard

This will install Tetra4D Automate on your computer.

It is recommended that you close all other applications before continuing.

Click Next to continue, or Cancel to exit Setup.

2. Read and accept the End User License Agreement



License Agreement
Please read the following important information before continuing.

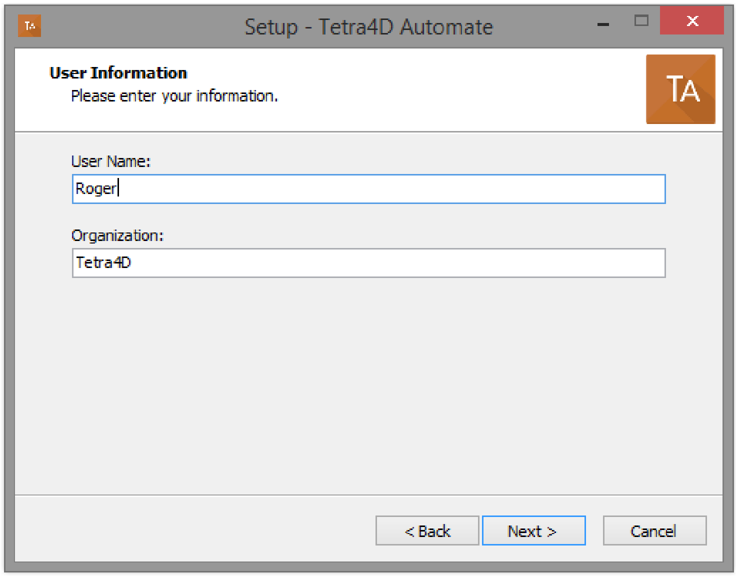
Please read the following License Agreement. You must accept the terms of this agreement before continuing with the installation.

End User License Agreement

NOTICE TO USER: PLEASE READ THIS AGREEMENT CAREFULLY. BY COPYING, INSTALLING OR USING ALL OR ANY PORTION OF THE SOFTWARE YOU ACCEPT ALL THE TERMS AND CONDITIONS OF THIS AGREEMENT, INCLUDING, IN PARTICULAR THE LIMITATIONS ON: USE CONTAINED IN SECTION 2; TRANSFERABILITY IN SECTION 4; WARRANTY IN SECTIONS 6 AND 7; LIABILITY IN SECTION 8; CONNECTIVITY AND PRIVACY IN SECTION 14; AND SPECIFIC PROVISIONS AND EXCEPTIONS IN SECTION 15. YOU AGREE THAT THIS AGREEMENT IS LIKE ANY WRITTEN NEGOTIATED AGREEMENT SIGNED BY YOU. THIS AGREEMENT IS ENFORCEABLE AGAINST YOU AND ANY LEGAL ENTITY THAT

[✓ I accept the agreement] I do not accept the agreement

3. Enter your user information (optional), then click **Next** to continue:



User Information
Please enter your information.

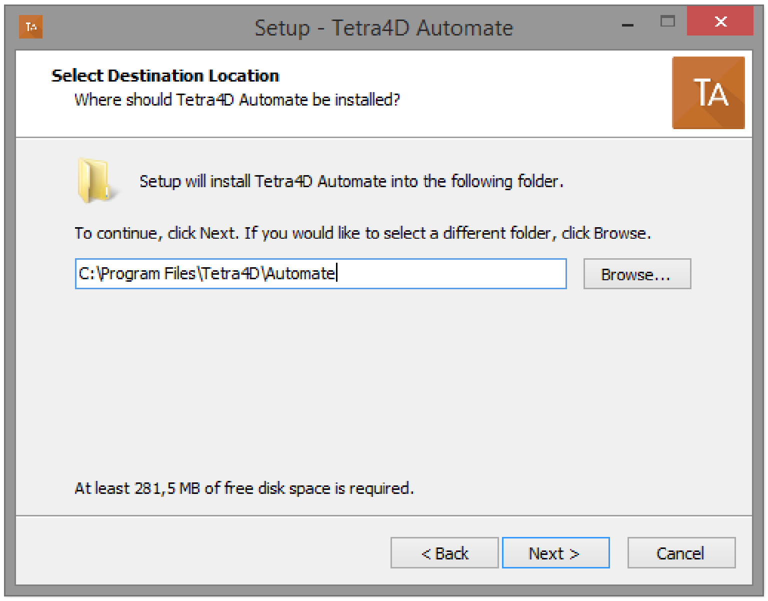
User Name: Roger

Organization: Tetra4D

4. Select the installation folder, then click **Next** to continue:

   Remark:

   The installation process will create the main folder and a set of sub-folders, including a sample data set that can be used to for deployment.



Setup - Tetra4D Automate

Select Destination Location

Where should Tetra4D Automate be installed?

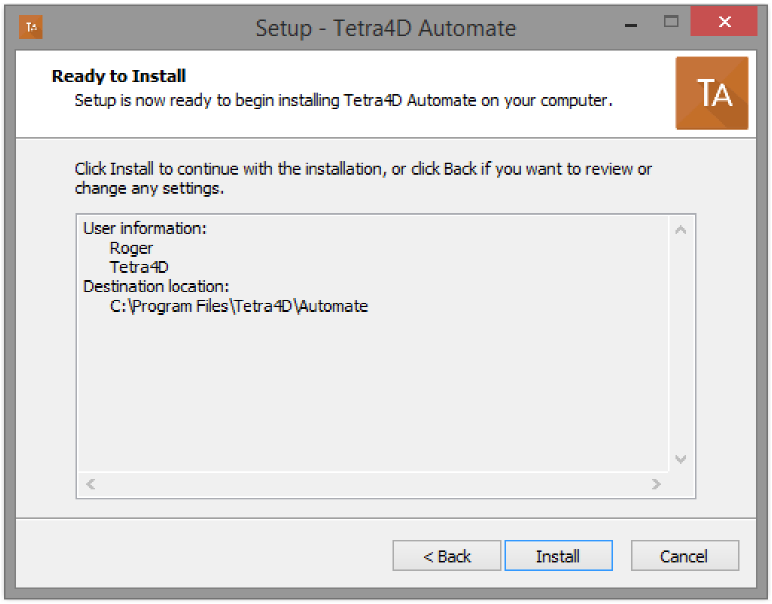
Setup will install Tetra4D Automate into the following folder.

To continue, click Next. If you would like to select a different folder, click Browse.

C:\Program Files\Tetra4D\Automate

At least 281,5 MB of free disk space is required.

5. Review the installation information and click **Install**:



Setup - Tetra4D Automate

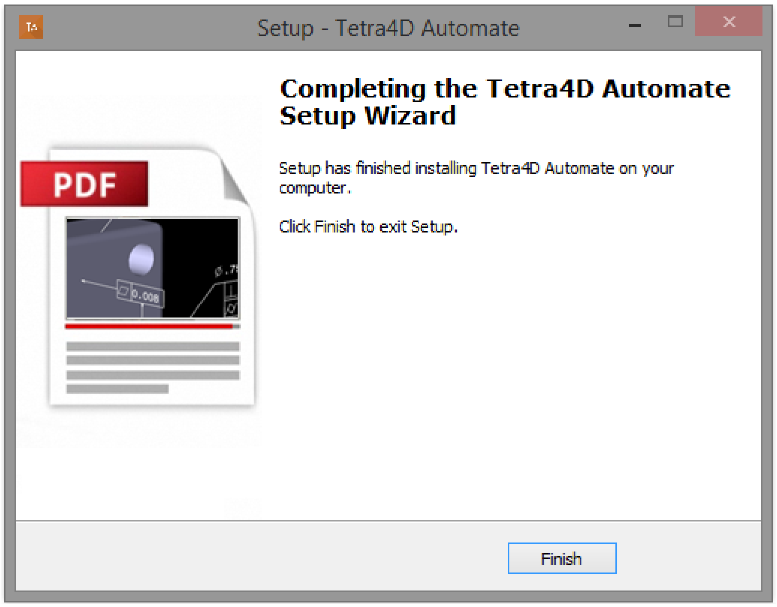
Ready to Install

Setup is now ready to begin installing Tetra4D Automate on your computer.

Click Install to continue with the installation, or click Back if you want to review or change any settings.

User information:
Roger
Tetra4D
Destination location:
C:\Program Files\Tetra4D\Automate

6. At the end of the installation process, click **Finish**.


Setup - Tetra4D Automate

Completing the Tetra4D Automate Setup Wizard

Setup has finished installing Tetra4D Automate on your computer.

Click Finish to exit Setup.

Installation folder structure

The installer creates the below folder structure inside the main installation folder:

• Install:	Installation information
• Resource:	Program resources
• Samples:	Sample data set (folders, CAD data, Conversion settings, Command line…)

Presentation of the Samples folder

Please refer to the Appendix 1 for detailed explanations about samples.

Activation and Registration

The Tetra4D Automate license is based on two elements that will be provided by the Tetra4D support:

- A serial number,
- A “Module” file that defines the available CAD readers and optional writers.

  Remark:

  The module file is tied to the serial number.
  Make it sure that the Module file you are using corresponds to the Serial number.
  To confirm, open the Module file and check the line “TetraLicence”.

The Tetra4D Automate licensing system makes it possible to activate your license on-line if the computer is connected to the Internet, or off-line if the computer is not connected to the Internet.

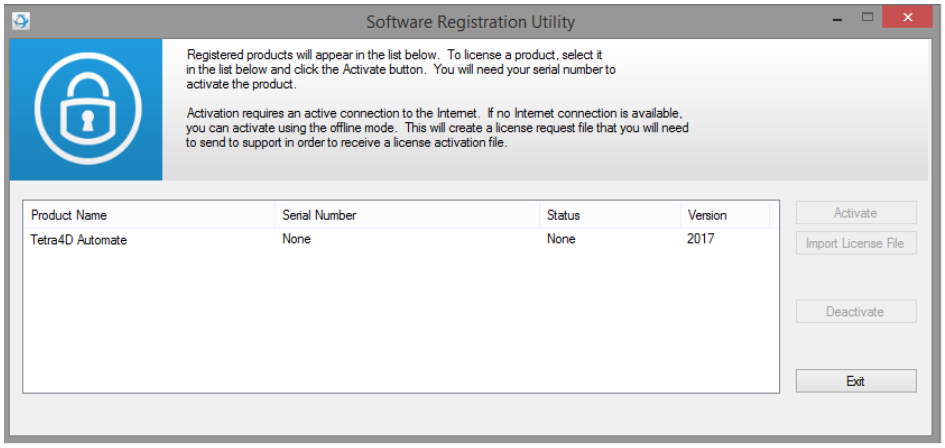
License Activation (on-line)

The procedure to activate your license is described below.

To access to the activation dialog:

- Open a Windows Explorer
- Open the folder where the Tetra4D Automate has been installed
- Run startLicenseManager.bat

  This batch program starts the Registration utility program



Registered products will appear in the list below. To license a product, select it in the list below and click the Activate button. You will need your serial number to activate the product.

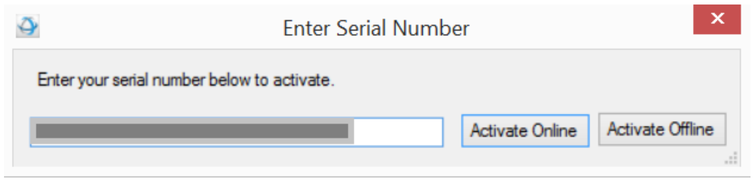
Activation requires an active connection to the Internet. If no Internet connection is available, you can activate using the offline mode. This will create a license request file that you will need to send to support in order to receive a license activation file.

Product Name	Serial Number	Status	Version
Tetra4D Automate	None	None	2017

Notes:

- The Tetra4D License Manager tool lists:
  - *The installed Tetra4D products*
  - *The status of each product (Trial mode, registered…)*
  - *The version of each product*
- Select the **Tetra4D Automate** product

- Click on **Activate**
- Input your serial number



Enter Serial Number

Enter your serial number below to activate.

- Click on Activate Offline
- Click **Exit** to close the window.
- Copy the "Module" file that has been delivered along your serial number into **C:\ProgramData\TechSoft3D\License**

License Activation (off-line)

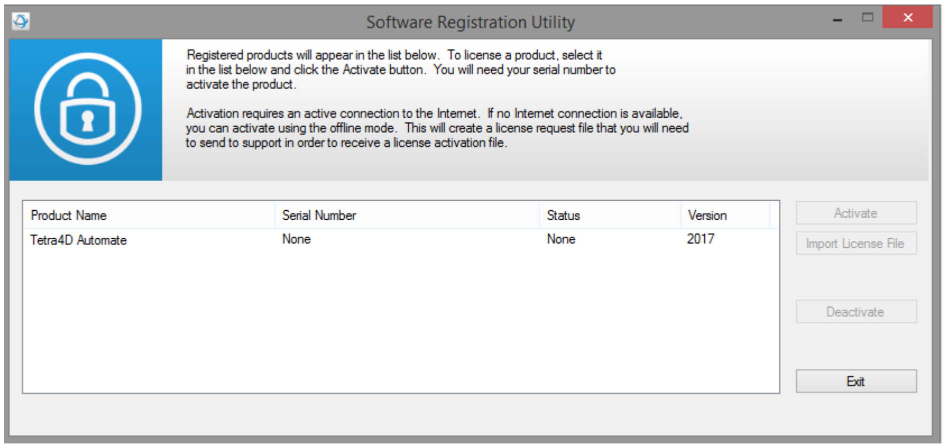
Tetra4D Automate enables an off-line activation of the license.

The procedure to activate your license is described below.

To access to the activation dialog:

- Open a Windows Explorer
- Open the folder where the Tetra4D Automate has been installed
- Run startLicenseManager.bat

  This batch program starts the Registration utility program



Software Registration Utility

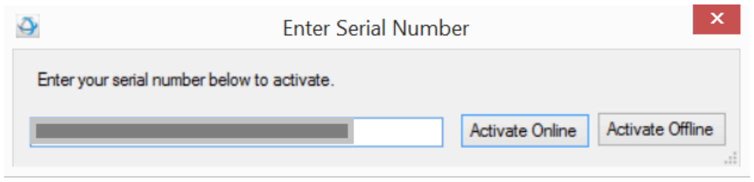
Registered products will appear in the list below. To license a product, select it in the list below and click the Activate button. You will need your serial number to activate the product.

Activation requires an active connection to the Internet. If no Internet connection is available, you can activate using the offline mode. This will create a license request file that you will need to send to support in order to receive a license activation file.

Product Name	Serial Number	Status	Version
Tetra4D Automate	None	None	2017

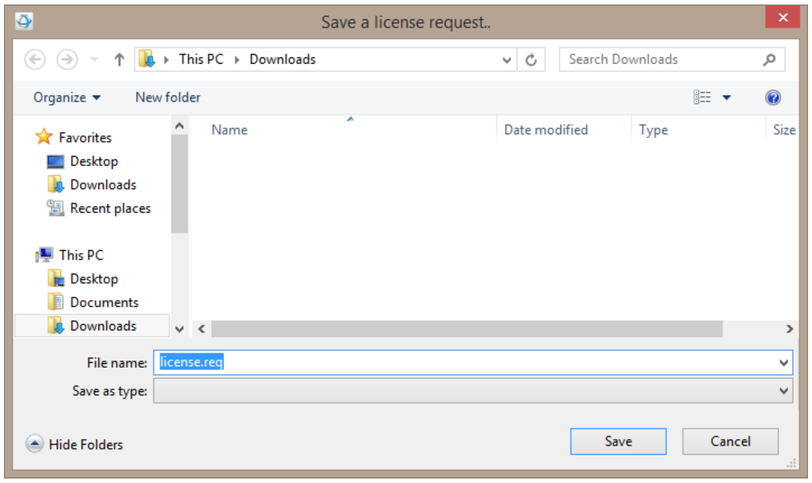
Notes:

- The Tetra4D License Manager tool lists:
    - *The installed Tetra4D products*
    - *The status of each product (Trial mode, registered…)*
    - *The version of each product*
- Select the **Tetra4D Automate** product
- Click on **Activate**
- Input your serial number



Enter your serial number below to activate.

- Click on Activate Offline
- Save the Offline_activation_ XXXXXXX-XXXXXXXXXX-XXXXXXX-XXXXX _macAddress.req file generated by the license manager



File name: license.req

Save as type:

- Send the Offline_activation_ XXXXXXX-XXXXXXXXXX-XXXXXXX-XXXXX _macAddress.req file to support@tetra4d.com
- Click **Exit** to close the window.

- Our support staff will send 2 license files corresponding to the computer from which you issued the activation request

  Remark:

- The files you will receive will be named as shown below:
- XXXXXXX-XXXXXXXXXX-XXXXXXX-XXXXX _macAddress.license
- TetraAutomate2017Test.modules

- Copy the 2 delivered files into C:\ProgramData\TechSoft3D\License
- Click **Exit** to close the window.

How to use Tetra4D Automate

Conversion process

During the conversion of the CAD files into a PDF document by **Tetra4D Automate**, the following operations are performed:

- The CAD data are read according to the specified conversion settings XML file,
- The 3D object from the specified PDF template is replaced by the read CAD data,
- The external information defined by XML files if any (attributes, title block) are read
- The Tetra4D Enrich widgets present in the template (Carousel of Views, Table) are updated based on the new 3D data and the read XML attributes file,
- The generic actions (text output on part selection, change of rendering mode…), if any, are updated,
- The text fields (title block) are updated based on the specified title block XML file if any, and on the new 3D data.

Method to trigger a conversion

Along with the source CAD files to be converted into PDF, it is required to prepare the following files to trigger a conversion.

PDF Template file (required)

Tetra4D Automate requires a template created by Tetra4D Enrich.

The PDF document used as a template must be created using **Tetra4D Enrich,** and must contain at least a 3D annotation.

Conversion Settings XML file (required)

XML file containing various settings for converting 3D file to PDF such as background color, importing element, etc. Please refer [Conversion Settings XML file] in details.

Parts attributes definition XML file (optional)

XML file containing part attributes. If some of the meta-data to be put in the PDF document are managed by another application (such as ERP, PLM…), so not present in the CAD files, Tetra 4D Automate enables to import them through an XML file and map them to the 3D.

Please refer [Parts attributes definition XML file] in details.

Text fields values definition XML file (optional)

XML file defining text information to populate text fields (title-block) in the PDF document.

Please refer [Text fields values definition XML file] in details.

Table definition XML file (optional)

XML file defining a table (having multiple rows, columns, and a title row). If the contents of the table to be shown in the PDF document is managed by another application (such as ERP, PLM…), Tetra 4D Automate enables to import it using an XML file, and to map the rows of the table with the 3D.

Please refer [Table definition XML file] in details.

Running Tetra4D Automate

Tetra4D Automate can be triggered thanks to a command line. Along with the Tetra4D Automate application path, the following arguments can be specified:

#	Identifier	Type of file	Required
(1)	-pdfin	Tetra 4D Enrich PDF template	**Yes if (2) isn't specified**
(2)	-htmlin	HTML template used to export 3D to HTML	**Yes if (1) isn't specified**
(3)	-libdir	Installation folder (where the A3DLIBS.dll is located)	No
(4)	-3dfilein	Source 3D file (1)	**Yes if (12) isn't a xml file and (1) is specified**
(5)	-3doptions	XML file: Conversion settings	**Yes**
(6)	-xmlattribsin	XML file: Import of meta-data	No
(7)	-xmldatafieldsin	XML file: Values to populate text fields	No
(8)	-xmlimport	XML file: Definition of a table	No
(9)	-attachfile or -attachlist	File to be attached in the PDF, or Text file containing a list of files to be attached	No
(10)	-outfpwdopen	Specification of a password that will be required to open the PDF document	No
(11)	-outfpwdright	Specification of a password that will be required to change the document permission settings	No
(12)	-outfname	Output file name and format (2)	**Yes**
(13)	-outlog	Log files	No

Remarks:

(1) Tetra4D Automate supports the same CAD formats as Tetra4D Converter
The CAD formats that are available for with one installation of the product are specified by the license provided with the Serial Number.

(2) Tetra4D Automate exports 3D PDF and HTML files. The ability to export to other CAD formats like STEP, IGES… is available optionally.
The availability of the optional export to a neutral format is specified by the license provided with the Serial Number.

Example of command line:

```
"C:\Automate.exe"^
 -pdfin "C:\Template_Automate_PartList.pdf"^
 -3dfilein "C:\_micro engine.CATProduct"^
 -xmldatafieldsin "C:\MicroEngine_Import_TITLEBLOCK.xml"^
 -xmlattribsin "C:\MicroEngine_Import_ATTRIB.xml"^
 -3doptions "C:\automate_settings_manual.xml"^
 -outfname "C:\Tetra4D_Micro_Engine_PartList.pdf"^
 -outlog "C:\Tetra4D_Micro_Engine_PartList.log"
```

Conversion settings

Four samples of conversion settings files are provided with the setup of tetra4D Automate.

Please refer to the Appendix 1 for detailed explanations about these samples.

Conversion setting file structure

The structure of the settings is based on three main sections:

- CAD files Import parameters,
- Export parameters,
- PDF parameters.

The different settings and options are presented below.

CAD files import Parameters

This section provides all the CAD data reading parameters and options that are meant to define what CAD entities are considered during the CAD data reading phase.

General

Specifies what information are considered during the reading phase.

Attribute	Description	Value
ReadSolids	Import solids	true: Import false: Not imported
ReadSurfaces	Import surfaces	Same as above
ReadWireframes	Import wireframes	Same as above
ReadFeature	Import feature	Same as above
ReadPmis	Import PMI and views	Same as above
ReadAttributes	Import Attributes	Same as above
ReadHiddenObjects	Import hidden objects	Same as above
ReadConstructionAndReferences	Import construction and references	Same as above
ReadActiveFilter	Import active filter	Same as above
ReadDrawings	Import drawings	Same as above
ReadGeomTessMode	Import geometry (B-rep) and/or polygon (tessellation)	0: Geometry only 1: Both geometry and tessellation 2: Tessellation only
DefaultUnit	Default Unit	0: Point 1: Inch 2: Millimeter 3: Centimeter 4: Picas 5: Foot 6: Yard 7: Meter 8: Kilometer 9: Mile 10: Unknown
LoadStructureOnly	Load only the assembly structure	**true: Not supported** false: Import the full assembly

VIEW filtering

Specifies if the native CAD views have to be filtered.

Remarks:

- The views filtering feature applies to the following CAD formats only:
    - Inventor,
    - CATIA V5.

Attribute	Description	Value
IgnoreSubAsmViews	Specifies if the views that are defined in the sub-levels and the components of an assembly are retained in the PDF document or if they are ignored	true: All views from sub-levels are ignored false: All views are read
FilterViewsByName	Enables to define a rule to filter the views according to their name Mode: Defines the filtering mode Criteria: Defines the string of character the will be used to identify the views	Mode: - 0: No Filtering - 1: Views matching with criteria are kept - 2: Views matching with criteria are ignored Criteria: String of characters

PMI

Specifies how the PMIs will be displayed in the PDF document.

Attribute	Description	Value
AlwaysSubstituteFont	Specifies if the PMI font defined in the CAD is used or substituted by another one	true: Substitutes the font false: Use the CAD font
SubstitutionFontValue	Substitution font	Name of font
NumberOfDigitsAfterDot	Number of decimal places to use for numeric values if no decimal information is specified in the CAD file	0 ~ 5
DefaultUnit	Units of measure of the PMI to use if no unit information is specified in the CAD file	0: Point 1: Inch 2: Millimeter 3: Centimeter 4: Picas 5: Foot 6: Yard 7: Meter 8: Kilometer 9: Mile 10: Unknown
AlwaysUseDefaultColor	"true" to substitute the color of PMI defined in the CAD file by the new color specified in <DefaultColor>	true: Yes false: No

ProprietaryFontDirectories

The fonts are usually retrieved from system folders. These paths enable to specify additional locations for fonts.

DefaultColor

If AlwaysUseDefaultColor is set to “true”, the DefaultColor key defines the color of the PMIs. The color is defined with RGB values:

- Black: Red="0" Green="0" Blue="0"
- White: Red="1" Green="1" Blue="1"

Tessellation

Specifies the quality of the tessellation.

Remarks:

- The tessellation attributes are defined in the “Import Parameters” section because tessellation is calculated during the reading of the CAD data.
- Note that depending on the CAD format, the tessellation might be read directly from the native CAD data.

Attribute	Description	Value
TessellationLevelOfDetail	Quality of tessellation	0: Extra low 1: Low 2: Medium 3: High 4: Extra high 5: User defined 6: Controlled precision
ChordHeightRatio	Chord height ratio	50 ~ 10000
AngleToleranceDeg	Angle tolerance degrees	10 ~ 40
MinimalTriangleAngleDeg	Minimal triangle degrees	0.01 ~ 2147483648.00
MaxChordHeight	Max chord height	1 ~ 30
Accurate	Accurate Tesselation or not	true: Yes false: No
KeepUVPoint	Specifies if we want to keep the UV points	**true: Not supported** false: No
UseHeightInsteadOfRatio	Use Height instead of Ratio	true: Yes false: No
MaximalTriangleEdgeLength	Maximal edges’ length of the triangle in tesselation	

Remark:

If TesselationLevelOfDetail is set in between 0 to 4 (included), all the other values are ignored.
If TesselationLevelOfDetail is set to 5 or 6, then the other attributes must be defined.

Assembly

Defines the CAD files reading strategy and enable to specify additional folders to look for child parts if they are not found in the main assembly folder or sub-folders.

Attribute	Description	Value
UseRootDirectory	Use the root directory to look for sub-assemblies and child parts	true: Yes false: No
RootDirRecursive	Look for sub-assemblies and child parts into the root directory sub-folders recursively	true: Yes false: No

SearchDirectories

Enables to specify additional search directories.

PathDefinitions

Enables to specify additional search directories by selecting a text file.

Multiple Entries

Enables to specify the configuration to be read if the file contains different configurations.

Incremental loading

The only supported value is “false”.

Specific

Enables to define some format specific parameters.

Export Parameters

This section provides settings that define the format of the document to be created:

- Choice of the export format,
- Definition of the export options for the selected format.

Export settings

The A3DExportSettings value specifies the export format.

Note that the 3D PDF (value 0) is the default format and that all the other formats are optionally accessible.

Attribute	Description	Value
RemoveExtension	Removes the extension	true: Yes false: No
PDF (PRC)		0
STEP		1
IGES		2
JT		3
3MF		4
STL		5
VRML		6
PARASOLID		7
HTML		1000

Export to PDF (PRC)

Defines the compression options for the export to 3D PDF format.

Attribute	Description	Value
CompressBrep	Activates the compression of the Brep or not	true: Yes false: No
CompressTesselation	Activates the compression of the tesselation or not	true: Yes false: No
CompressBrepTol	Specifies the tolerance for the Brep compression	0.01 to 0.1

Export to other CAD formats (Step, Iges…)

Defines the export options for the selected export format.

Export to HTML Format

Convert a 3D PDF file in interactive HTML files.
To visualize the HTML output file in your browser, see "Description of the script: 50_start_ConvertToHTML.bat".

Security settings

Defines the export options for the selected export format.

Attribute	Description	Value
Printing	Enables print feature and defines printing quality	true: High quality print Low: low quality print false: No printing
Edit	Enables the Edit feature	true: Yes false: No
Comments	Enables the Comment feature	true: Yes false: No
DocAssembly	Enables edit the document (page insertion, merging)	true: Yes false: No
Fill and sign	Enables to fill fields and to use the Sign feature	true: Yes false: No
Copy	Enables to copy the contents of the document	true: Yes false: No
Secure	Enables to modify the security settings of the document	true: Yes false: No
SaveAs	Enables to "save as" the document	true: Yes false: No
Templates	Enables to use the document as a template	true: Yes false: No

Remarks:

- Some of these options may prevent the document's JavaScript to run.

PDF Parameters

This section provides parameters that apply only for the export to 3D PDF.

Colors

This section enables to define an index of colors that can be referenced by other settings.
Remark:

    The colors are defined by RGB values.
    The colors can be selected in the index based on the order
    First color is 0

Example:

```
<ColorData Red="1" Green="1" Blue="1" />          : White
<ColorData Red="0" Green="0" Blue="0" />          : Black
<ColorData Red="0" Green="1" Blue="0" />          : Green
<ColorData Red="0.5" Green="0.5" Blue="0.5" /> …
```

RectData

Will be documented in a future version of the document

Artwork Data

These settings are related to display of PMIs and the orientation of the default view.

Attribute	Description	Value
JavaSriptFileName	Not used	
AnimationStyle	Not used	
ActivatePMICrossHiglight	Activate PMIs Cross Highlight	true: yes false: no
AddPMISemanticInformation	Add information about PMIs semantic	true: yes false: no
ChangePMIColor	Specifies if we want to change the PMI's color	true: yes, set to PMIColorIndex false: no
PMIColorIndex	Color of the markup of the artwork if ChangePMIColor=true	Index in list color
AddStandardViews	Enable the creation of 6 standard views: Left, Top, Front Right, Bottom, Back in Orthographic projection mode	true: yes false: no
DefaultViewOrientation	Specifies which type of default view's orientation we want to use	0: Default Acrobat 1: Default CatiaV5 2: Default Creo 3: Default Inventor Inch 4: Default Inventor Flat 5: Default Inventor mm 6: Default Nx 7: Default SolidEdge 8: Default SolidWorks 9: User defined
DefaultViewVerticalAxis	Definition of the vertical axis to be used in the default view if DefaultViewOrientation=9	0: X Axis 1: Y AXis 2: Z AXis
DefaultViewVerticalRotation	Rotation angle of the model on vertical axis	0-360
DefaultViewHorizontalRotation	Rotation angle of the model on horizontal axis	0-360

AnnotData

These settings are related to the 3D annotation:

- Background color and lighting
- Rendering mode
- Data tree and 3D toolbar visualization…

Attribute	Description	Value
OpenModelTree	Specifies if we want the model tree to be open when the 3D annot is activated	true: yes false: no
ShowToolbar	Specifies if we want the 3d toolbar to be show	true: yes false: no
Lighting	Default light	0: Current light 1: No light 2: White light 3: Day light 4: Bright light 5: Primary color light 6: Night light 7: Blue light 8: Red light 9: Cube light 10: CAD optimized light 11: Headlamp light
RenderingStyle	Default rendering style	0: Transparent bounding box 1: Solid 2: Transparent 3: Solid wireframe 4: Illustration 5: Solid outline 6: Shaded illustration 7: Bounding box 8: Transparent bounding box outline 9: Wireframe= 10: Shaded wireframe 11: Transparent wireframe

		12: Hidden wireframe 13: Vertices 14: Shaded vertices
BackgroundColorIndex	Default background color	0 ~ Index of color specified in Colors (Top is 0)
TransparentBackground	Specifies if the 3D annotation has a transparent background False recommended	true: yes false: no
ActivateWhen	Specifies when the 3d annotation will be activated	0: remain inactive until explicitly activated by a script or user action. 1: be activated as soon as the page containing the 3D Annot is opened. 2: be activated as soon as any part of the page containing the 3D Annot becomes visible.
DesactiveWhen	Specifies when the 3d annotation will be deactivated	0: remain active until explicitly activated by a script or user action. 1: be deactivated as soon as the page containing the 3D Annot is opened. 2: be deactivated as soon as any part of the page containing the 3D Annot becomes invisible.
AppearanceBorderWidth	Border width in points	0 ~
PosterImageIndex	Not documented	
ArtWorkIndex	Not documented	

Description of the files used to import additional data

Tetra4D Automate provides the ability to import additional data into the PDF document to complement the CAD data:

- Meta-data,
- Text information to populate text fields,
- Part list.

XML file to import meta-data

XML schema of files that enable the import of meta-data.

```
<?xml version="1.0"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
 <xsd:element name="attributes">
  <xsd:complexType>
   <xsd:sequence>
    <xsd:element name="NAME">
     <xsd:complexType>
      <xsd:sequence>
       <xsd:element name="NEW_ATTRIB">
        <xsd:complexType>
         <xsd:attribute name="name" type="xsd:string" use="required" />
         <xsd:attribute name="value" type="xsd:string" use="required" />
        </xsd:complexType>
       </xsd:element>
      </xsd:sequence>
      <xsd:attribute name="key" type="xsd:string" use="required" />
     </xsd:complexType>
    </xsd:element>
   </xsd:sequence>
  </xsd:complexType>
 </xsd:element>
</xsd:schema>
```

Multiple attributes & values can be specified for every key

Sample of an XML file where several attributes (SUPPLIER, PRICE, SUPPLIER_REFERENCE) are imported and mapped to the 3D parts, using in that case the name of the part as mapping key.

```
<attributes>
<NAME key="HEAD_TUBE">
       <NEW_ATTRIB name="SUPPLIER" value="Sup_1" />
       <NEW_ATTRIB name="PRICE" value="111,93 €" />
       <NEW_ATTRIB name="SUPPLIER_REFERENCE" value="9397608" />
</NAME>
</attributes>
```

XML file to define values to populate text fields

The following is XML schema of data field XML file.

```
<?xml version="1.0"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
 <xsd:element name="Fields">
  <xsd:complexType>
   <xsd:sequence>
    <xsd:field name="name" value=” value” />
    <xsd:field name="name" value=” value” />
   </xsd:sequence>
  </xsd:complexType>
 </xsd:element>
</xsd:schema>
```

Each field has one value.

Sample of an XML file where several values to populate text fields are defined.

The mapping key are project, description, revision, and must be defined in the template using the Tetra4D Enrich feature “Set populate method for text fields).

```
<?xml version="1.0"?>
<Fields>
	<Field key="project" value="Micro Engine" />
	<Field key="description" value="Spare Parts catalog" />
	<Field key="revision" value="2016/02/29" />
</Fields>
```

XML file to define a table

Sample of an XML file that enable to import a part list in a template.

The information are:

- **Coldefs**: Definition of the names of the columns of the table.
  Remark:
  The Coldefs values enable to map the table with the parts from the 3D annotation.
  In such a case, the selection of a row in the table will highlight the mapped part in the 3D annotation and conversely, the selection of a part in the 3D annotation will highlights the mapped row in the table.

- **Tr Is-header**=true: Definition of the titles of the columns.
  Remark:
  This option enables to replace the attributes names defined by the Coldefs by a text labels.

- Tr: Definition of the values for the rows of the table.

```
<TS3DFullXml CurrentVersionFormat="401">
        <TS3DBoms>
                <BOMTable>
                        <coldefs>
                                <col attributename="ItemNumber" is-key="true" is-visible="true"/>
                                <col attributename="SUPPLIER"/>
                                <col attributename="SUPPLIER_REFERENCE"/>
                        </coldefs>
                        <tr is-header="true">
                                <td>Item</td>
                                <td>Supplier</td>
                                <td>Reference</td>
                        </tr>
                        <tr>
                                <td>A4123589</td>
                                <td>Supplier A</td>
                                <td>RefA01</td>
                        </tr>
                </BOMTable>
        </TS3DBoms>
</TS3DFullXml>
```

Mapping an imported table with 3D parts

The mapping of the imported table with the 3D parts is based on the values of one specified CAD attribute. To enable the mapping, the table must contain a column that lists the values of the attribute used as the mapping key.

The definition of the column used to map the table with the components of the 3D annotation is done by the `is-key` value.
In addition to the mapping key definition, the `is-visible` value makes it possible to hide one column (usually the one used for the mapping) if it is not relevant to show these values in the table.

Example:

```
              <coldefs>
                     <col attributename="ItemNumber" is-key="true" is-visible="true"/>
                     <col attributename="SUPPLIER"/>
                     <col attributename="SUPPLIER_REFERENCE"/>
              </coldefs>
```

- Above, the attribute ItemNumber is the mapping key (`is-key="true"`)
- The corresponding column will be visible (`is-visible="true"`)

Errors management

The Tetra4D "Automate" provides 3 different types of information to handle the conversion process.

- The Tetra4D "Automate" executable returns execution codes,
- 2 log files are generated during the conversion:
    - Log_File_Name.Log: Short log file
    - Log_File_Name_verb.Log: Verbose log file

Remark:

- The name of the log file "Log_File_Name.Log" is defined in the command line.
- The "_verb" suffix is automatically added by Tetra4D "Automate".

Short log file

The following is the sample of a short log file:

```
Application                    : Tetra4D Automate - 2017(2017.1.2) 05-09 23:48:37
-------------------------------
--- Command Line --------------
-------------------------------
C:\Program Files\Tetra4D\Automate\samples\Mode_Manual\\..\..\Automate.exe
-pdfin: C:\Program
       Files\Tetra4D\Automate\samples\Mode_Manual\\..\..\samples\templates\Template_Automate_ViewCompo
       nent.pdf
-libdir: C:\Program Files\Tetra4D\Automate\samples\Mode_Manual\\..\..\.
       -3dfilein: C:\Program
       Files\Tetra4D\Automate\samples\Mode_Manual\\..\..\samples\Mode_Manual\01_Components\housing
       front.CATPart
-xmldatafieldsin: C:\Program
       Files\Tetra4D\Automate\samples\Mode_Manual\\..\..\samples\Mode_Manual\01_Components\HousingFron
       t_Import_TITLEBLOCK.xml
-3doptions: C:\Program
       Files\Tetra4D\Automate\samples\Mode_Manual\\..\..\samples\settings\automate_settings_manual.xml
-outfname: C:\Program
       Files\Tetra4D\Automate\samples\Mode_Manual\\..\..\samples\Mode_Manual\10_PDF_Out\Tetra4D_Housin
       gFront_ViewComponent.pdf
-outlog: C:\Program
       Files\Tetra4D\Automate\samples\Mode_Manual\\..\..\samples\Mode_Manual\10_PDF_Out\Tetra4D_Housin
       gFront_ViewComponent.log
-------------------------------
--- Log Summary ---------------
-------------------------------
Date                           : 06/01/17
Time                           : 11:54:06
Reading data from file         : C:\Program
Files\Tetra4D\Automate\samples\Mode_Manual\\..\..\samples\Mode_Manual\01_Components\housing
front.CATPart
CPU time (sec)                 : 9.607
-------------------------------
COMMAND_ERRORS                 : 0
COMMAND_WARNINGS               : 0
-------------------------------
READING_ERRORS                 : 0
READING_WARNINGS               : 0
-------------------------------
WRITING_ERRORS                 : 0
WRITING_WARNINGS               : 0
-------------------------------
```

The short log file presents a summary of the conversion process:

- Application

  Version and date of the Tetra4D “Automate” software

- Command line

  Copy of the command line and arguments that have triggered the conversion

- Log Summary

  Date and time of the conversion (start)

  Name of the file that has been converted

  Duration of the conversion

- Command errors and warnings

  Errors and warnings related to the command line

  Remark:

  > The errors that can be identified are:
  >
  > - Syntax errors in command line
  > - Missing file (a specified file has not been found)
  > - Permissions errors (ie: Not possible to write in the output folder)
  >
  > The conversion settings defined in the conversion settings XML file are not validated during that stage.

- Reading errors and warnings

  Errors and warnings issued during the CAD data reading stage.

- Writing errors and warnings

  Errors and warnings issued during the PDF writing stage.

See “Tetra4D “Automate” errors” to access to a list of managed errors

Verbose log file

The Verbose log file provides detailed information about the conversion process.

See “Tetra4D “Automate” errors” to access to a list of managed errors

Tetra4D “Automate” errors

The below information correspond to the warning and errors that are listed in the LOG files.
They are formatted as follows:

[type] [code] - [message]

Type:

The possible values for TYPE are:

- WAR

  Warning

- ERR

  Error

Code and Message:

- 3 : missing parameter in the command line
- 4 : can't find a file from the command line
- 5 : can't initialize library
- 6 : can't load settings file
- 18 : the maximum converter instance number on the server has been reached
- -34 : License option doesn't allowed the requested function
- -205 : Cannot initialize library
- -1000000 : Unexpected PDFLib Error.
- -1000002 : A 3D is required to save the document.
- -1000003 : document has not been saved properly.
- -1000005 : file is already opened in another application.
- -1000006 : file is locked by another application.
- -1000009 : Cannot access file.
- -1000011 : Cannot find a default view in 3D Annotation.
- -1000015 : Not enough memory.
- -1000016 : The font is not supported.
- -1000020 : In a PRC/PDF file, there is a view without a camera; snapshots cannot be created from such views.
- -1000021 : Tabletopdf or one of its components is missing from your directory.
- -1000022 : Failed to create the table

- -1000023 : Failed to create the table because of style entry
- -1000024 : Resource directory could not be initialized
- -1000026 : Error while generating the images for the buttons in the view carousel.
- -1000028 :   Failed to create the table
- -1000029 : Error in text data
- -1000030 : PDFlib DLL or one of its components is missing.
- -1000031 : ImageMagick DLL or one of its components is missing.
- -1000032 : Error retrieving 3D data on page. No 3D annotation was found.
- -1000033 : Error retrieving 3D data on page. No 3D annotation was found.
- -1000035 : Error reading text stream.
- -1000036 : Encoding error converting text stream.
- -1000037 : Error in slider creation
- -1000038 : Error in slider creation
- -1000039 : The slidetable object could not be found
- -1000040 : The carousel object could not be found
- -1000041 : The XML contained into the PDF is not compatible with the library

CAD readers errors

The below information corresponds to the warning and errors that are listed in the LOG files.
They are formatted as follows:

[type] [code base format]:[IO parameter]:[base code]:[specific code] - [message]

Type

The possible values for TYPE value are:

- WAR

  Warning
- ERR

  Error

Code Base Format

- CGR
- CV5
- PROE
- UG
- U3D
- SLW
- SLW
- CAT
- IDEAS
- CADDS
- STEP
- IGES
- INV
- SAT
- XVL
- SE
- JT
- XT : Parasolid
- VDA

I/O Parameters

The possible values for Input/Output value are:

- R

  Reader

- W

  Writer

Parasolid Specific Errors:

The error information is defined by a “Base Code” and a “Specific Code” values
The Base Code possible values are:

- -1011 : Reader failure

  For this Base Code, the Specific Code values can be:

- 202: Error in surface creation
- 218: Failed to keep parsed entities

- -1099 : Unknown

  For this Base Code, the Specific Code values can be:

- 201: Unknown density units

Tetra4D “Automate” return codes

The executable returns a code that enable to detect if the process was successful or not.
Managing the return code is a first step in handling the conversion process.

Return code: 0

This value shows that the executable was correctly triggered:

- Command line is correct
- License was successfully read
- Conversion settings file has been successfully read

  Remark:

  In this version, the values that are defined in the conversion settings file are not checked.

Remark:

Even if the return code is 0, some warning or errors may have occurred during the PDF generation process. For this reason, the LOG files (SHORT and VERBOSE) must be parsed in order to access to these additional information.

Return code: Positive value (3, 4, 5, 6, 18)

These values show errors that are related to the trigger of the “Automate” executable:

- 3 : missing parameter in the command line
- 4 : can't find a file from the command line
- 5 : can't initialize library
- 6 : can't load settings file
- 18 : the maximum converter instance number on the server has been reached

Remark:

These errors stop the execution of the process.

These errors are also listed in the LOG files.

Return code: Negative value (-34)

This value is issued if there is a licensing issue (CAD reader license).

- -34 : License option doesn't allowed the requested function

----

APPENDIX 1: Presentation of samples

Prior to any installation, please check with your Tetra4D software provider the requirements and recommendations to run the Tetra4D "Automate".

Introduction

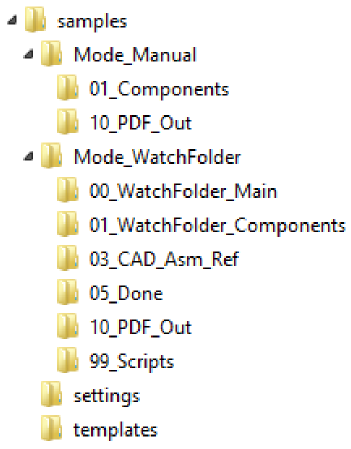
The Sample folder, its sub-folders and all the contained files are provided as samples of Tetra4D Automate implementations.
Remark:

Depending on the customer environment and on the deployment requirements, the implementation may differ.

This data set is meant to easily and quickly understand:

- how the software can be triggered,
    - Command line with the arguments,
    - Watched folder
- the formats of the different XML files
    - Conversion settings,
    - Import of external attributes

The SAMPLES folder is organized as follows:



```
samples
  Mode_Manual
    01_Components
    10_PDF_Out
  Mode_WatchFolder
    00_WatchFolder_Main
    01_WatchFolder_Components
    03_CAD_Asm_Ref
    05_Done
    10_PDF_Out
    99_Scripts
  settings
  templates
```

- **SAMPLES** (root folder)

  The root folder contains the sub-folders:
    - **Mode_Manual**
      Basic sample material to demonstrate the major features of Tetra4D Automate and how it can be triggered by a simple script.
    - **Mode_WatchFolder**
      Sample material to simulate a watched folder implementation where Tetra4D Automate is triggered by events occurring in the "watched folder".
    - **Settings**
      Folder containing the different conversion settings files used by the samples.
    - **Templates**
      Folder containing the different templates used by the samples.

“Manual mode” sample description

Sample organization

- **Mode_Manual** (root folder)

  The root folder contains:
  - **Files:**
    - 10_start_ViewComponent.bat
    - 20_start_PartList.bat
    - 30_start_PartList_ImportBOM.bat
    - 40_start_EngineeringDataRelease.bat
    - 50_start_ConvertToHTML.bat
  - **01_Components**
    Input folder where the CAD data to be Converted are stored.
  - **10_PDF_Out**
    Output folder where the PDF document and the Log files are written by Tetra4D Automate.
  - **20_HTML_Out**
    Output folder where the HTML, XML, server setting, and the Log files are written by Tetra4D Automate.

Description of the script: 10_start_ViewComponent.bat

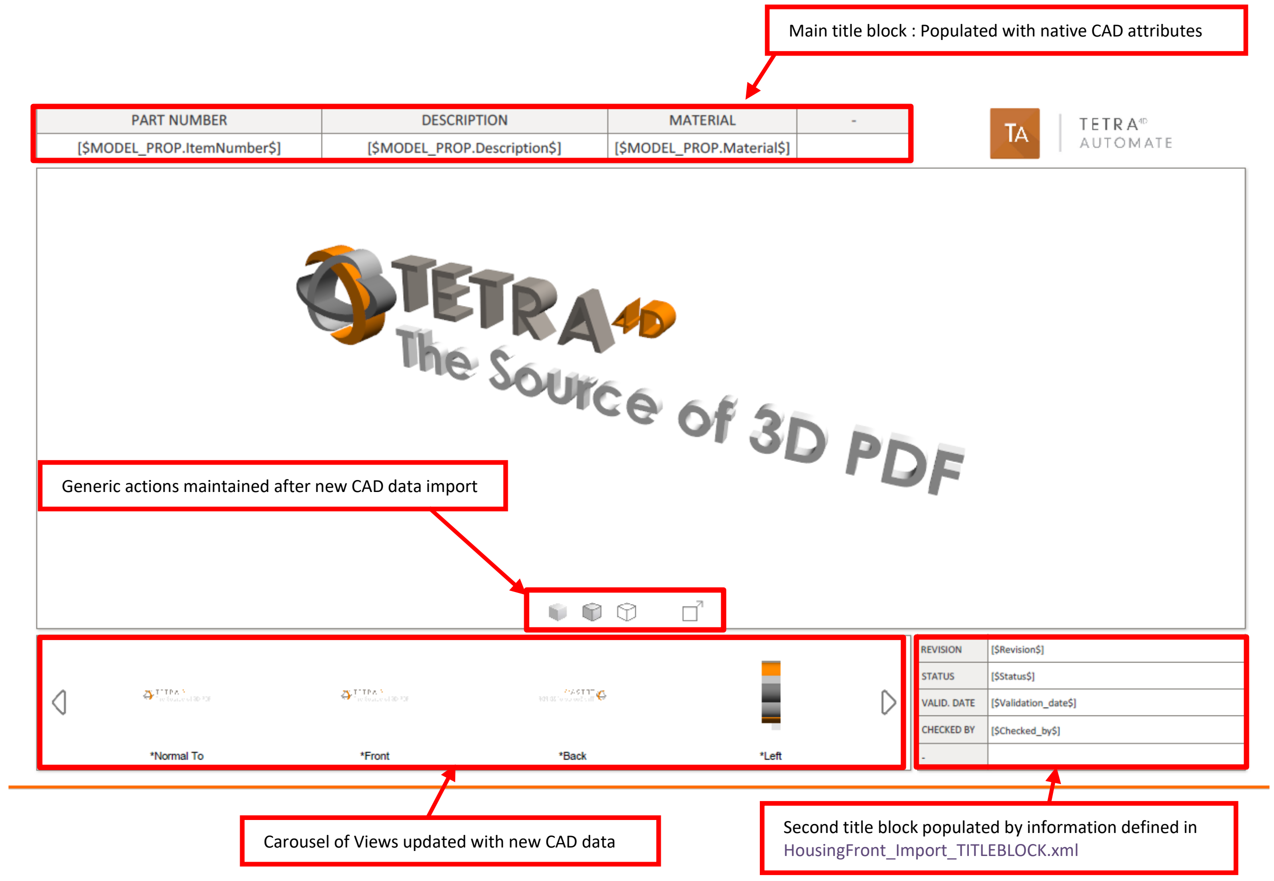
This sample illustrates the generation of a component visualization PDF document.

Script

This script triggers Tetra4D Automate with the following arguments:

```
"%ROOT_DIR%\Automate.exe"^
 -pdfin             "%TEMPLATES_DIR%\Template_Automate_ViewComponent.pdf"^
 -libdir            "%ROOT_DIR%\."^
 -3dfilein          "%CAD_DIR%\housing front.CATPart"^
 -xmldatafieldsin   "%CAD_DIR%\HousingFront_Import_TITLEBLOCK.xml"^
 -3doptions         "%SETTINGS_DIR%\automate_settings_manual.xml"^
 -outfname          "%OUT_DIR%\Tetra4D_HousingFront_ViewComponent.pdf"^
 -outlog            "%OUT_DIR%\Tetra4D_HousingFront_ViewComponent.log"
```

Template: Template_Automate_ViewComponent.pdf



Main title block : Populated with native CAD attributes

PART NUMBER	DESCRIPTION	MATERIAL	-
[$MODEL_PROP.ItemNumber$]	[$MODEL_PROP.Description$]	[$MODEL_PROP.Material$]	

Generic actions maintained after new CAD data import

REVISION	[$Revision$]
STATUS	[$Status$]
VALID. DATE	[$Validation_date$]
CHECKED BY	[$Checked_by$]
-	

Carousel of Views updated with new CAD data

Second title block populated by information defined in HousingFront_Import_TITLEBLOCK.xml

Running the sample

In Windows Explorer, double click on the script **10_start_ViewComponent.bat**

A command window appears during the conversion process and can be closed by the user by pressing any key, when the process will be ended.

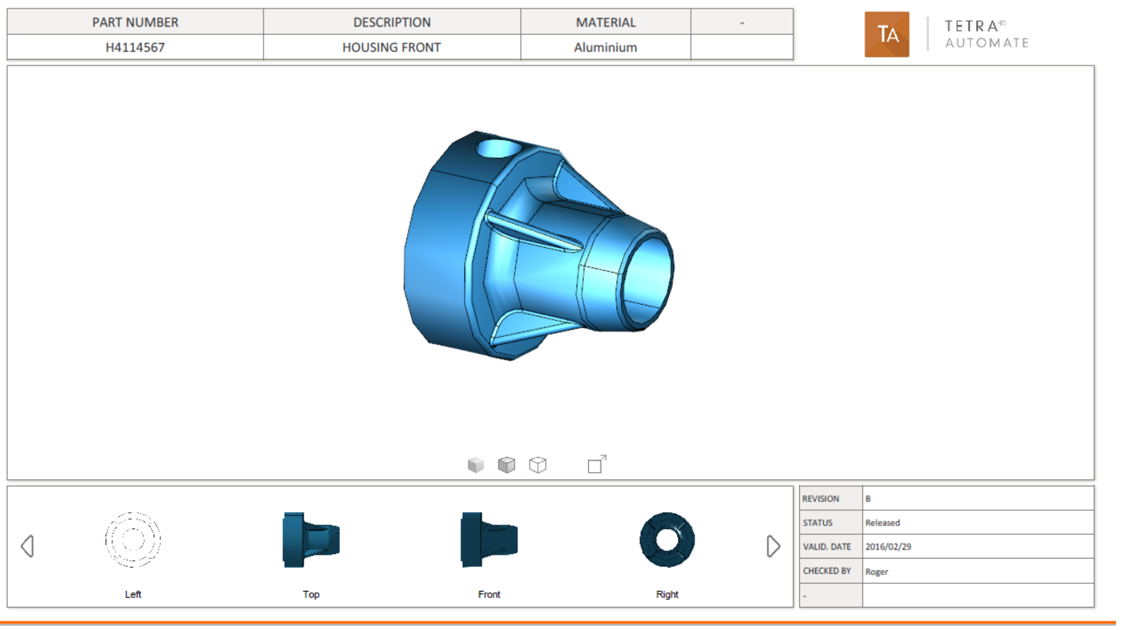
Open the folder **10_PDF_Out** to access to the results.

Remark:

3 files are created:

- PDF file
- Short log file
- Verbose log file.

Results:



PART NUMBER	DESCRIPTION	MATERIAL	-
H4114567	HOUSING FRONT	Aluminium	

REVISION	B
STATUS	Released
VALID. DATE	2016/02/29
CHECKED BY	Roger
-	

Description of the script: 20_Start_PartList.bat

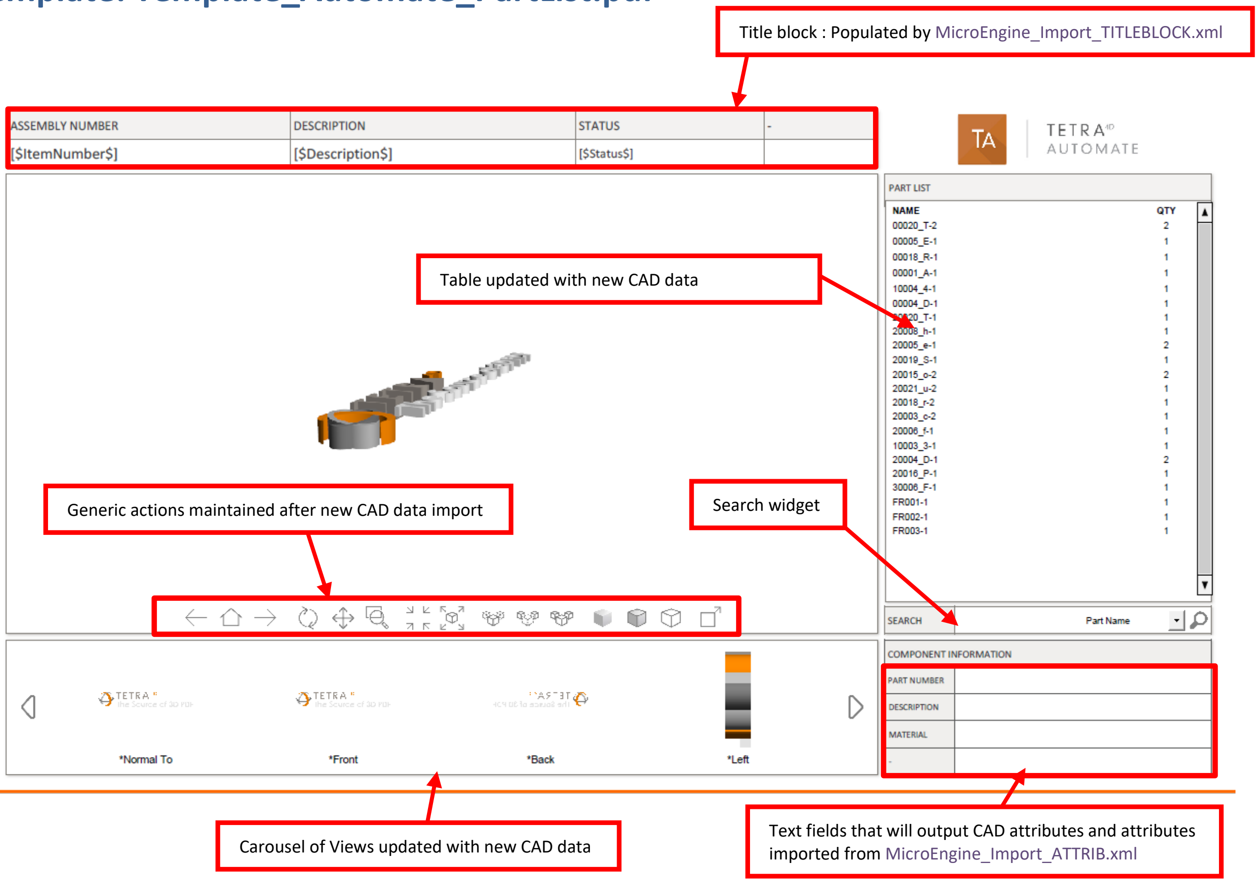
This sample illustrates the generation of a part list PDF document.

Script

This script triggers Tetra4D Automate with the following arguments:

```
"%ROOT_DIR%\Automate.exe"^
 -pdfin              "%TEMPLATES_DIR%\Template_Automate_PartList.pdf"^
 -libdir             "%ROOT_DIR%\."^
 -3dfilein           "%CAD_DIR%\_micro engine.CATProduct"^
 -xmldatafieldsin    "%CAD_DIR%\MicroEngine_Import_TITLEBLOCK.xml"^
 -xmlattribsin       "%CAD_DIR%\MicroEngine_Import_ATTRIB.xml"^
 -3doptions          "%SETTINGS_DIR%\automate_settings_manual.xml"^
 -outfname           "%OUT_DIR%\Tetra4D_Micro_Engine_PartList.pdf"^
 -outlog             "%OUT_DIR%\Tetra4D_Micro_Engine_PartList.log"
```

Template: Template_Automate_PartList.pdf



Title block : Populated by MicroEngine_Import_TITLEBLOCK.xml
Table updated with new CAD data
Generic actions maintained after new CAD data import
Search widget
Carousel of Views updated with new CAD data
Text fields that will output CAD attributes and attributes imported from MicroEngine_Import_ATTRIB.xml
TETRA AUTOMATE
*Normal To
*Front
*Back
*Left

ASSEMBLY NUMBER	DESCRIPTION	STATUS	-
[$ItemNumber$]	[$Description$]	[$Status$]	

PART LIST

NAME	QTY
00020_T-2	2
00005_E-1	1
00018_R-1	1
00001_A-1	1
10004_4-1	1
00004_D-1	1
20020_T-1	1
20008_h-1	1
20005_e-1	2
20019_S-1	1
20015_o-2	2
20021_u-2	1
20018_r-2	1
20003_c-2	1
20006_f-1	1
10003_3-1	1
20004_D-1	2
20016_P-1	1
30006_F-1	1
FR001-1	1
FR002-1	1
FR003-1	1

SEARCH Part Name

COMPONENT INFORMATION

PART NUMBER	
DESCRIPTION	
MATERIAL	
-	

Running the sample

In Windows Explorer, double click on the script **20_start_PartList.bat**

A command window appears during the conversion process and can be closed by the user by pressing any key, when the process will be ended.

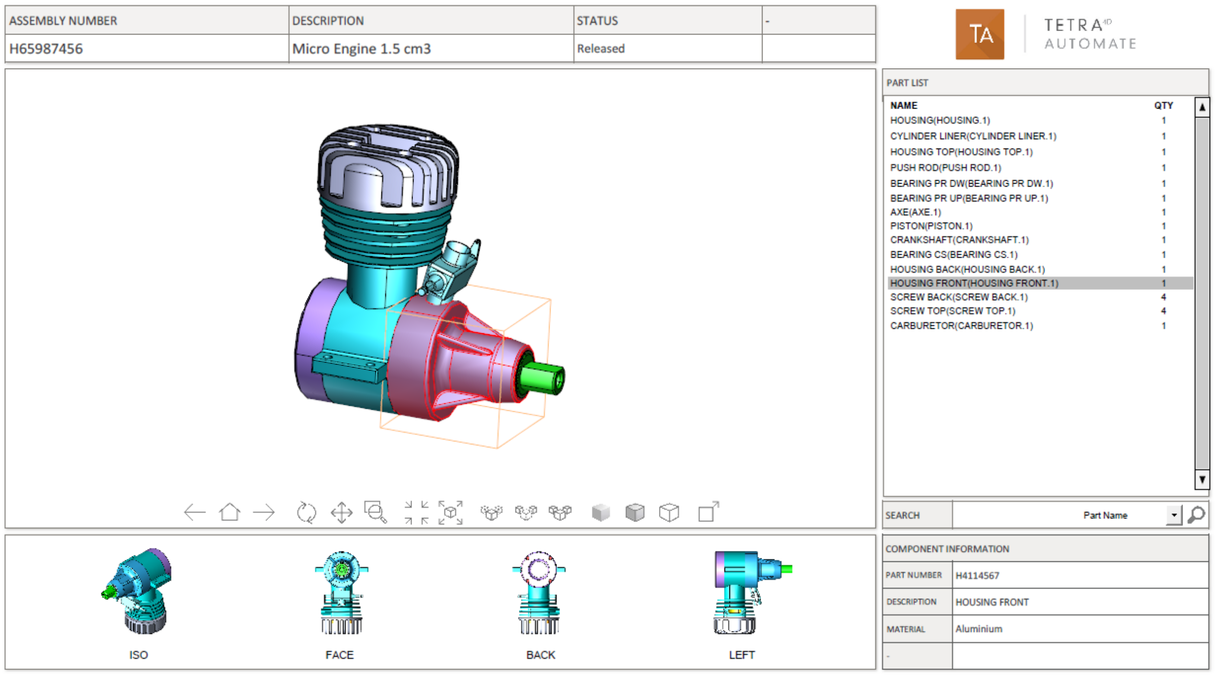
Open the folder **10_PDF_Out** to access to the results.

Remark:

3 files are created:

- PDF file
- Short log file
- Verbose log file.

Results:



ASSEMBLY NUMBER	DESCRIPTION	STATUS	-
H65987456	Micro Engine 1.5 cm3	Released	

PART LIST

NAME	QTY
HOUSING(HOUSING.1)	1
CYLINDER LINER(CYLINDER LINER.1)	1
HOUSING TOP(HOUSING TOP.1)	1
PUSH ROD(PUSH ROD.1)	1
BEARING PR DW(BEARING PR DW.1)	1
BEARING PR UP(BEARING PR UP.1)	1
AXE(AXE.1)	1
PISTON(PISTON.1)	1
CRANKSHAFT(CRANKSHAFT.1)	1
BEARING CS(BEARING CS.1)	1
HOUSING BACK(HOUSING BACK.1)	1
HOUSING FRONT(HOUSING FRONT.1)	1
SCREW BACK(SCREW BACK.1)	4
SCREW TOP(SCREW TOP.1)	4
CARBURETOR(CARBURETOR.1)	1

SEARCH: Part Name

COMPONENT INFORMATION

PART NUMBER	H4114567
DESCRIPTION	HOUSING FRONT
MATERIAL	Aluminium
-	

Description of the script: 30_Start_PartList_ImportBOM.bat

This sample illustrates the generation of a part list PDF document having an imported part (XML file) list instead of a calculated one.

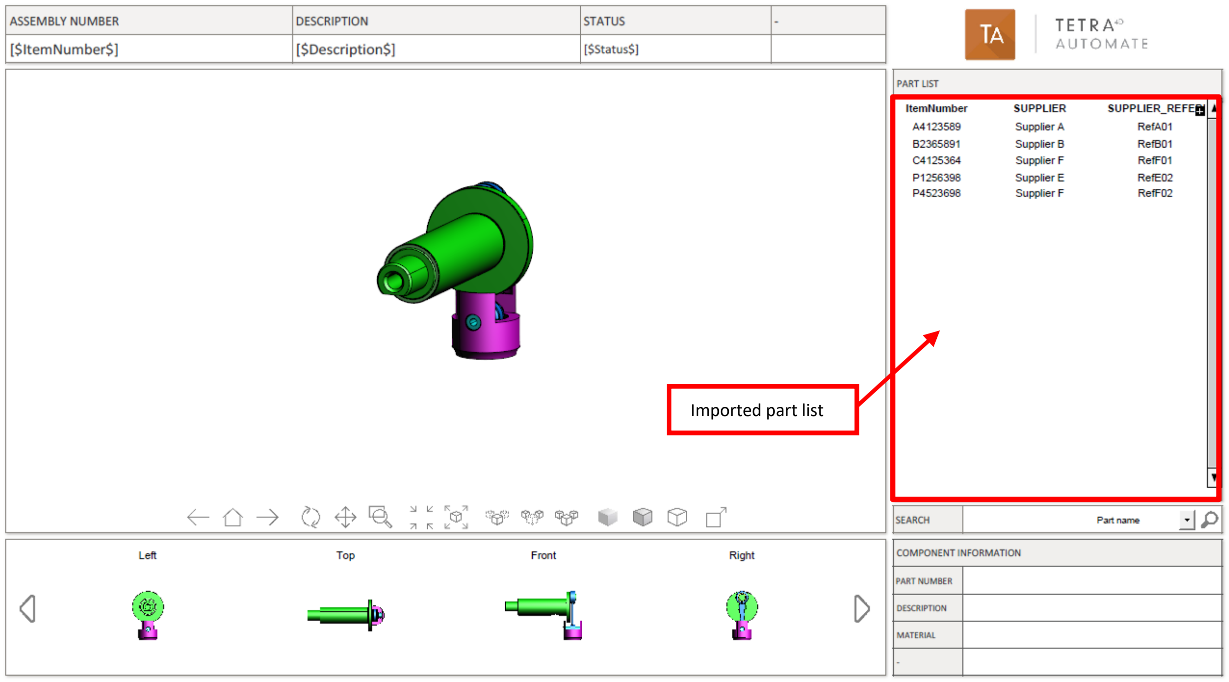
Script

This script triggers Tetra4D Automate with the following arguments:

```
"%ROOT_DIR%\Automate.exe"^
 -pdfin             "%TEMPLATES_DIR%\Template_Automate_PartList_InportBOM.pdf"^
 -libdir            "%ROOT_DIR%\."^
 -3dfilein          "%CAD_DIR%\_micro engine.CATProduct"^
 -xmldatafieldsin   "%CAD_DIR%\MicroEngine_Import_TITLEBLOCK.xml"^
 -xmlattribsin      "%CAD_DIR%\MicroEngine_Import_ATTRIB.xml"^
 -xmlimport         "%CAD_DIR%\BOM_Import.xml"^
 -3doptions         "%SETTINGS_DIR%\automate_settings_manual.xml"^
 -outfname          "%OUT_DIR%\Tetra4D_Micro_Engine_PartList_BOM.pdf"^
 -outlog            "%OUT_DIR%\Tetra4D_Micro_Engine_PartList_BOM.log"
```

Template: Template_Automate_PartList_ImportBOM.pdf

This template is similar to the one from previous sample, except for the part list that has been imported and mapped with the 3D annotation (this operation is mandatory to enable the update of template by Automate with a BOM import).



ASSEMBLY NUMBER	DESCRIPTION	STATUS	-
[$ItemNumber$]	[$Description$]	[$Status$]	

PART LIST

ItemNumber	SUPPLIER	SUPPLIER_REFE
A4123589	Supplier A	RefA01
B2365891	Supplier B	RefB01
C4125364	Supplier F	RefF01
P1256398	Supplier E	RefE02
P4523698	Supplier F	RefF02

COMPONENT INFORMATION

PART NUMBER	
DESCRIPTION	
MATERIAL	
-	

Running the sample

In Windows Explorer, double click on the script **30_start_PartList_ImportBOM.bat**

A command window appears during the conversion process and can be closed by the user by pressing any key, when the process will be ended.

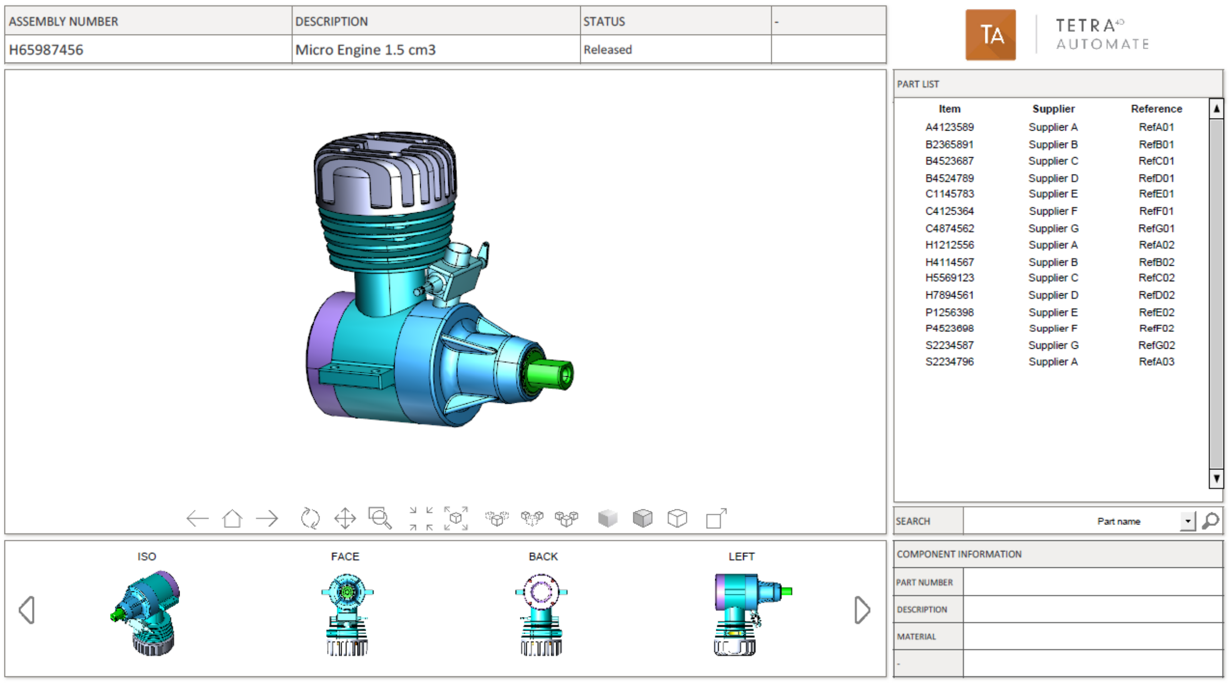
Open the folder **10_PDF_Out** to access to the results.

Remark:

3 files are created:

- PDF file
- Short log file
- Verbose log file.

Results:



ASSEMBLY NUMBER	DESCRIPTION	STATUS	-
H65987456	Micro Engine 1.5 cm3	Released	

PART LIST

Item	Supplier	Reference
A4123589	Supplier A	RefA01
B2365891	Supplier B	RefB01
B4523687	Supplier C	RefC01
B4524789	Supplier D	RefD01
C1145783	Supplier E	RefE01
C4125364	Supplier F	RefF01
C4874562	Supplier G	RefG01
H1212556	Supplier A	RefA02
H4114567	Supplier B	RefB02
H5569123	Supplier C	RefC02
H7894561	Supplier D	RefD02
P1256398	Supplier E	RefE02
P4523698	Supplier F	RefF02
S2234587	Supplier G	RefG02
S2234796	Supplier A	RefA03

COMPONENT INFORMATION

PART NUMBER	
DESCRIPTION	
MATERIAL	
-	

Description of the script: 40_EngineeringDataRelease.bat

This sample illustrates the generation of a “Technical data package” PDF document having an attached STP file.

This sample is based on two conversions done by Tetra4D Automate:

- First conversion: CAD file into a STEP file
- Second conversion: CAD file into a PDF document, with attachment of the existing STEP file.

Remark:

The export feature to neutral formats such as STEP, IGES… is an option of Tetra4D Automate.

Script

The script triggers 2 conversions:

<u>Conversion into STEP</u>

```
"%ROOT_DIR%\Automate.exe"^
 -pdfin             "%TEMPLATES_DIR%\Template_Automate_EngineeringDataRelease.pdf"^
 -3dfilein          "%CAD_DIR%\BracketMachined.catpart"^
 -3doptions         "%SETTINGS_DIR%\automate_settings_manual_ExportStep.xml"^
 -outfname          "%OUT_DIR%\Tetra4D_BracketMachined.stp"^
 -outlog            "%OUT_DIR%\Tetra4D_EngineeringDataRelease_STP.log"
```

<u>Conversion into PDF</u>

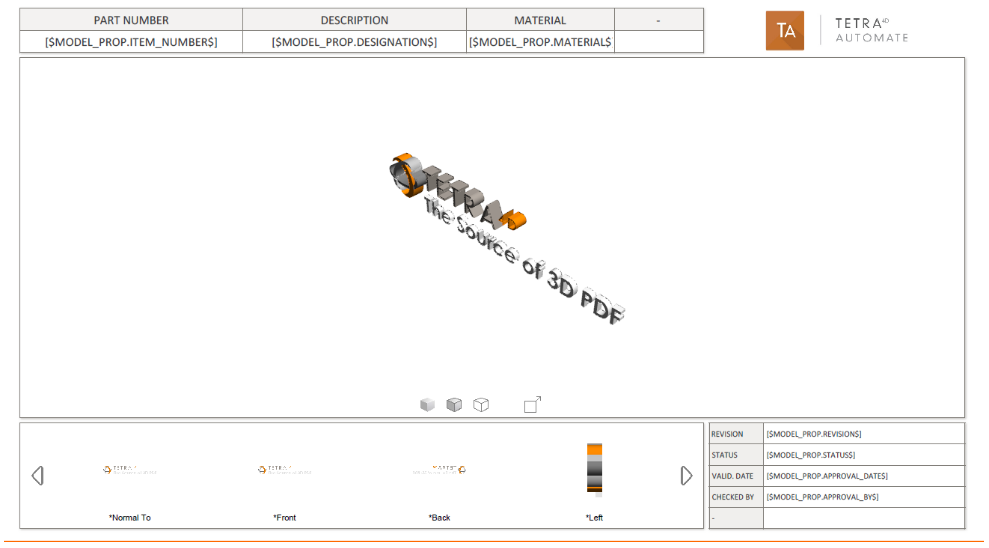
```
"%ROOT_DIR%\Automate.exe"^
 -pdfin             "%TEMPLATES_DIR%\Template_Automate_EngineeringDataRelease.pdf"^
 -libdir            "%ROOT_DIR%\."^
 -3dfilein          "%CAD_DIR%\BracketMachined.catpart"^
 -3doptions         "%SETTINGS_DIR%\automate_settings_manual.xml"^
 -attachfile        "%OUT_DIR%\Tetra4D_BracketMachined.stp"^
 -outfname          "%OUT_DIR%\Tetra4D_EngineeringDataRelease.pdf"^
 -outlog            "%OUT_DIR%\Tetra4D_EngineeringDataRelease.log"
```

Remark:

A template must be specified during the export to neutral format, even if it won’t be used.
The nature of the export format is specified in the conversion settings XML file.

Template: Template_Automate_PartList_ImportBOM.pdf

This template is similar to the one used with the first sample.



PART NUMBER	DESCRIPTION	MATERIAL	-
[$MODEL_PROP.ITEM_NUMBER$]	[$MODEL_PROP.DESIGNATION$]	[$MODEL_PROP.MATERIAL$]	

REVISION	[$MODEL_PROP.REVISION$]
STATUS	[$MODEL_PROP.STATUS$]
VALID. DATE	[$MODEL_PROP.APPROVAL_DATE$]
CHECKED BY	[$MODEL_PROP.APPROVAL_BY$]
-	

Running the sample

In Windows Explorer, double click on the script **40_start_EngineeringDataRelease.bat**

A command window appears during the conversion process and can be closed by the user by pressing any key, when the process will be ended.

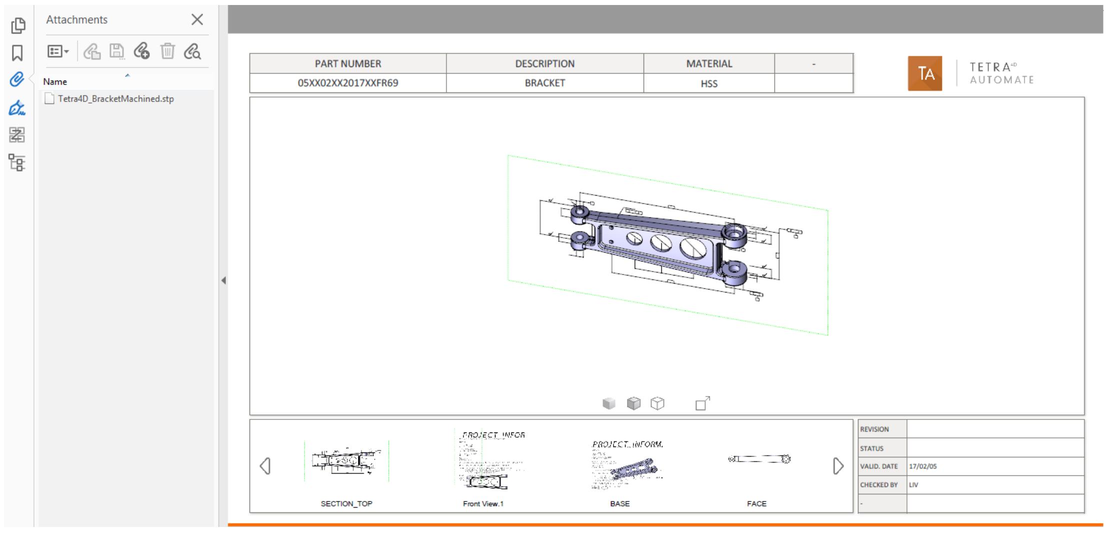
Open the folder **10_PDF_Out** to access to the results.

Remark:

6 files are created:

- STEP file, Short log file for STEP, Verbose log file for STEP
- PDF file, Short log file for PDF, Verbose log file for PDF

Results:



PART NUMBER	DESCRIPTION	MATERIAL	-
05XX02XX2017XXFR69	BRACKET	HSS	

REVISION	
STATUS	
VALID. DATE	17/02/05
CHECKED BY	LIV
-	

Description of the script: 50_start_ConvertToHTML.bat

This sample illustrates the generation of a HTML document from an existing 3D PDF, and how to visualize it.

Script

This script triggers Tetra4D Automate to convert a 3D PDF file into an HTML file. It also starts a client server process to open the default web browser with the URL to view the Html file:

```
"%ROOT_DIR%\Automate.exe"^
 -pdfin       "%TEMPLATES_DIR%\Template_Automate_PartList_ImportBOM.pdf"^
 -3doptions   "%SETTINGS_DIR%\automate_settings_manual.xml"^
 -outfname    "%OUT_DIR%\PdfTemplate_Automate_PartList_ImportBOM.xml"^
 -outlog      "%OUT_DIR%\PdfTemplate_Automate_PartList_ImportBOM.log"
start start_server.bat
start "" ^
"http://localhost:11180/viewer.html?sample=PdfTemplate_Automate_PartList_ImportBOM&modeltree=none&toolbar=none"
```

First step:

Triggers Tetra4D Automate to export HTML and the associated XML files from the PDF file in input:

`-pdfin`

The two generated files are put in the folder:

`samples\Mode_Manual\20_HTML_Out\root\PdfContent`

Second step:

Remark:

You need Python version 3.4.XX at least to be installed

You can retrieve it ***from the official Python download page****.*

Launches a client/server mode, which uses the 2 previous file XML and HTML generated.
The XML file is the entry point of all. It represents the PDF document. The XML and HTML files will be parsed and displayed in the browser by a client web viewer that is provided as part of Automate.
See the samples directory in the package download:

`samples\ samples\Mode_Manual\20_HTML_Out.`

To help testing this functionality, a simple HTTP server is provided, and can be launched by clicking on:

`samples\Mode_Manual\20_HTML_Out\start_server.bat.`

Third step:

Opens the following URL in your browser:

```
localhost:11180/viewer.html?sample=
PdfTemplate_Automate_PartList_ImportBOM
```

You can load your own 3D PDF converted into HTML by updating the sample query string parameter with the XML file name describing your document without the .xml file extension:

```
localhost:11180/viewer.html?sample=XmlFilenameWithoutExtension
```

Remark:

Be sure to properly URL-encode 'XmlFilenameWithoutExtension'.
For instance, there are no spaces in a URL, so any space character in your file name must be replaced by "%20" (without the quotes) for a space character.
See: ***https://www.w3schools.com/tags/ref_urlencode.asp*** for more information about URL encoding.

By default, the viewer of a 3D annotation will display a toolbar and a panel will display the model tree and views. You can hide them by adding the following to your query string:

`&modeltree`=none to hide the panel with the model tree and views.

`&toolbar`=none to hide the toolbar.

See the "Browser Compatibility" chapter to correctly open the generated html file.

Description of the script: 60_start_ConvertToHTML_Offline.bat Offline viewing

The python script `convert_offline.py` is provided to enable an offline consumption of the Html file.
It merges the generated XML/HTML/attachments files to remove cross file references.
This mode of consumption enables a lighter integration of the produced files.
For instance:

- Direct access on the intranet
- Simple mobile reading

Script

This script triggers Tetra4D Automate to convert a 3D PDF file into HTML for off-line consumption:

```
"%ROOT_DIR%\Automate.exe"^
 -pdfin       "%TEMPLATES_DIR%\Template_Automate_PartList_ImportBOM.pdf"^
 -3doptions   "%SETTINGS_DIR%\automate_settings_manual.xml"^
 -outfname    "%OUT_DIR%\PdfTemplate_Automate_PartList_ImportBOM.xml"^
 -outlog      "%OUT_DIR%\PdfTemplate_Automate_PartList_ImportBOM.log"
python convert_offline.py "%OUT_DIR%\PdfTemplate_Automate_PartList_ImportBOM.xml"
"%HTML_ROOT_DIR%" "%HTML_ROOT_DIR%\viewer.html" notoolbar nomodeltree
```

First step:

Triggers Tetra4D Automate to export HTML and the associated XML files from the PDF file in input:

`-pdfin`

The two generated files are put in the folder:

`samples\Mode_Manual\20_HTML_Out\root\PdfContent`

Second step:

Remark:

You need Python version 3.4.XX at least to be installed
You can retrieve it ***from the official Python download page****.*

Then from a command line tool, change to 20_HTML_Outdirectory and use the script as below:

```
python convert_offline.py LAYOUT_XML_FILE VIEWER_HTML_FILE OUTPUTDIR
[nomodeltree] [notoolbar]
```

Input files:

LAYOUT_XML_FILE:	Input layout XML file
VIEWER_HTML_FILE:	Input viewer.html file

Output:

OUTPUT_DIR: The output directory

This directory will contain the generated file. The file will be named after the XML file.

For instance, entering:

```
python convert_offline.py
root\PdfContent\PdfTemplate_Automate_PartList_ImportBOM.xml
root\ root\viewer.html notoolbar nomodeltree
```

Will result in the generation of `root\sample_DemoDataModel.html`

This Html file can be accessed by directly clicking on it.

Remark:

Be careful, even if the consumption can be done without any HTTP server, you need to keep the following folders and files installed on the computer where the Html is accessed:

`root\css`, `root\js`, `root\rsc`

and the file

`root\BrowserNotSupported.html`

to read:

`root\sample_DemoDataModel.html`.

Remark:

It is possible to remove the dependencies mentioned above so that the Html becomes self-containing and monolithic.

Please contact our support team for any information about this.

See the "Browser Compatibility" chapter to correctly open the generated html file.

Browser compatibility

The generated document can be viewed using:

- Microsoft Edge 19+
- Google Chrome 69+
- Safari 11.1+
- Firefox 60+

Description of the script: 70_Convert3DToHTML.bat

This sample illustrates the direct creation of a html file from a 3D CAD model and a HTML template.

The generated HTML file can be consumed without any HTTP server on all kind of browsers/devices.

Script

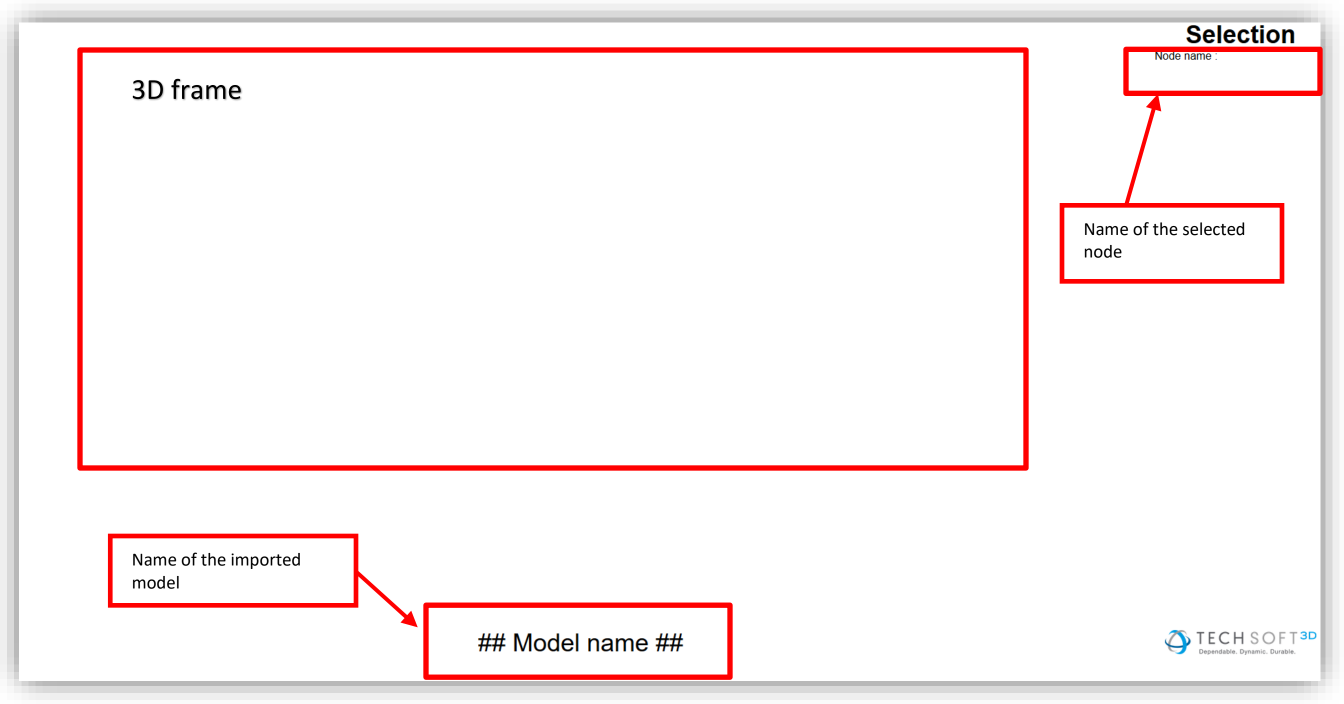
This script triggers Tetra4D Automate with the following arguments:

Conversion into STEP

```
"%ROOT_DIR%\Automate.exe"^
 -htmlin      "%TEMPLATES_DIR%\export3dtohtml_template_custom.html"^
 -3dfilein    "%CAD_DIR%\_micro engine.CATProduct"^
 -3doptions   "%SETTINGS_DIR%\automate_settings_manual_Export3DToHTML.xml"^
 -outfname    "%OUT_DIR%\Tetra4d_Micro_Engine.html"^
 -outlog      "%OUT_DIR%\Tetra4d_Micro_Engine.log"
```

Template: export3dtohtml_template_custom.html

This file is a basic template provided as an example.
Please contact our support team for any information about customization.



Running the sample

In Windows Explorer, double click on the script **_Convert3DToHTML.bat**

A command window appears during the conversion process and can be closed when the process ends by pressing any key.

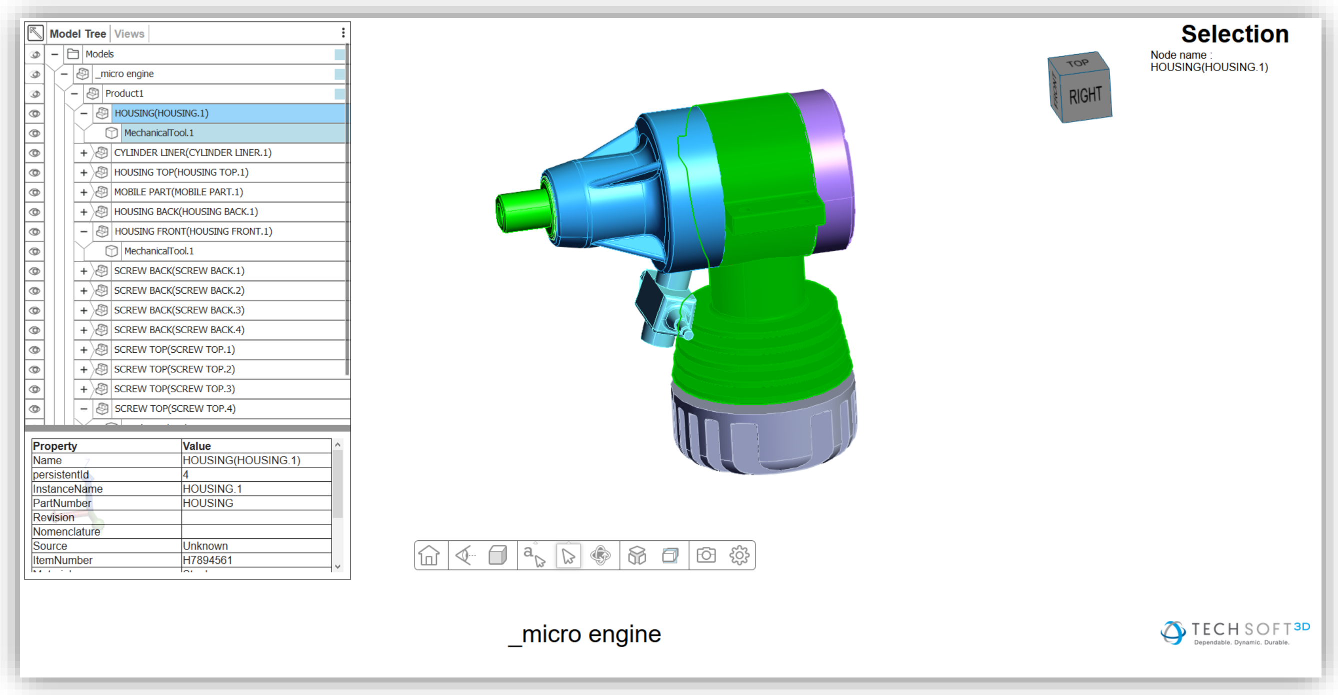
Open the folder **20_HTML_Out** to access to the results.

Remark:

3 files are created:

- Monolithic HTML file
- Short log file
- Verbose log file.

Results:



```
Models
  _micro engine
    Product1
      HOUSING(HOUSING.1)
        MechanicalTool.1
      CYLINDER LINER(CYLINDER LINER.1)
      HOUSING TOP(HOUSING TOP.1)
      MOBILE PART(MOBILE PART.1)
      HOUSING BACK(HOUSING BACK.1)
      HOUSING FRONT(HOUSING FRONT.1)
        MechanicalTool.1
      SCREW BACK(SCREW BACK.1)
      SCREW BACK(SCREW BACK.2)
      SCREW BACK(SCREW BACK.3)
      SCREW BACK(SCREW BACK.4)
      SCREW TOP(SCREW TOP.1)
      SCREW TOP(SCREW TOP.2)
      SCREW TOP(SCREW TOP.3)
      SCREW TOP(SCREW TOP.4)
```

Property	Value
Name	HOUSING(HOUSING.1)
persistentId	4
InstanceName	HOUSING.1
PartNumber	HOUSING
Revision	
Nomenclature	
Source	Unknown
ItemNumber	H7894561

Users can navigate the model tree, calculate physical properties, query user attributes, perform measurements, and navigate the View and PMI data.

Browser compatibility

The generated document can be viewed using:

- Microsoft Edge 19+
- Google Chrome 69+
- Safari 11.1+
- Firefox 60+

“Watch folder mode” sample description

Remark:

This sample is based on a basic folder organization and on simple scripts that enable to illustrate how Tetra4D Automate could be triggered by a “watch-folder” method (meaning that the generation is triggered when a file is copied in the Watch-folder).
The provided folder organization and scripts are not meant to be directly deployed in a production environment.
According to the customer requirements and environment, the Tetra4D Automate invocation method may differ.

This provided scripts are not supported by Tetra 4D.

Sample organization

- Mode_WatchFolder(root folder)

  The root folder contains:

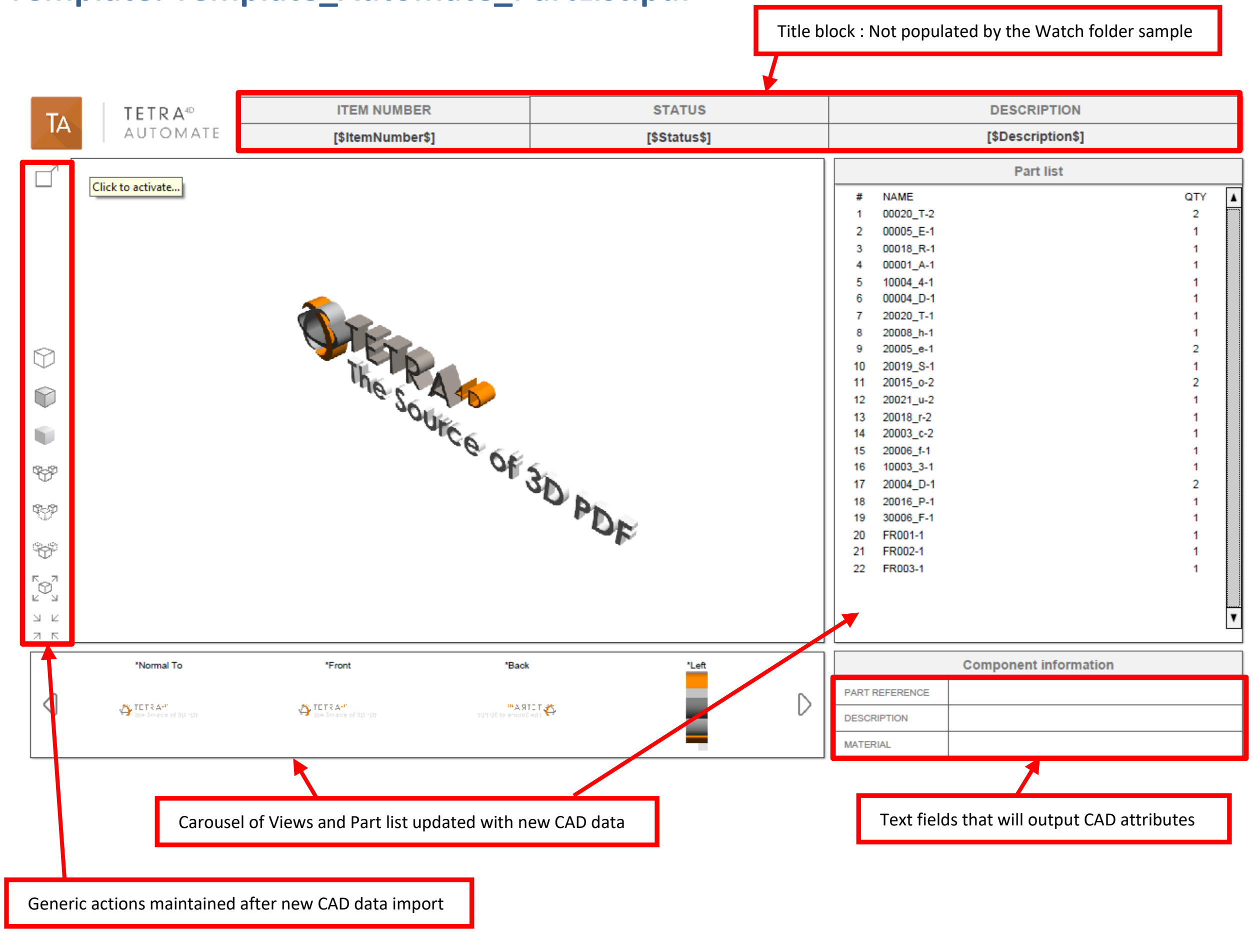
  - **Files:**
    - start_WatchFolderMode.bat
  - **00_WatchFolder_Main**
    Main watch folder.
  - **01_WatchFolder_Components**
    Components watch folder
  - **03_CAD_Asm_ref**
    Sample CAD assembly (full assembly)
  - **05_done**
    Folder where the processed files (from the Main watch folder) are copied after the conversion.
  - **10_PDF_Out**
    Conversion output folder where the PDF document and the Log files are written by Tetra4D Automate.
  - **99_Scripts**
    Scripts used to manage the watch-folder mechanism and to trigger the conversion in case of a new file to be converted.

Description of the sample

This sample has been designed to convert single components or assemblies into 3D PDF documents.
Remark:

> Since the conversion is based on a watch-folder mechanism (meaning that the end user will copy his files into the watch folder in order to trigger the conversion) the aggregation of meta-data from external XML file and the import of title block from an XML file haven't been implemented.
> However, it could be possible to modify the scripts in order to manage these additional XML files.

Template: Template_Automate_PartList.pdf



Title block : Not populated by the Watch folder sample

ITEM NUMBER	STATUS	DESCRIPTION
[$ItemNumber$]	[$Status$]	[$Description$]

Part list

#	NAME	QTY
1	00020_T-2	2
2	00005_E-1	1
3	00018_R-1	1
4	00001_A-1	1
5	10004_4-1	1
6	00004_D-1	1
7	20020_T-1	1
8	20008_h-1	1
9	20005_e-1	2
10	20019_S-1	1
11	20015_o-2	2
12	20021_u-2	1
13	20018_r-2	1
14	20003_c-2	1
15	20006_f-1	1
16	10003_3-1	1
17	20004_D-1	2
18	20016_P-1	1
19	30006_F-1	1
20	FR001-1	1
21	FR002-1	1
22	FR003-1	1

Component information

PART REFERENCE	
DESCRIPTION	
MATERIAL	

Carousel of Views and Part list updated with new CAD data

Text fields that will output CAD attributes

Generic actions maintained after new CAD data import

Running the sample

In Windows Explorer, double click on the script **start_WatchFolderMode.bat**

A command window appears, running the script **start_WatchFolderMode.vbs**

Remark:

The **start_WatchFolderMode.vbs** script manages the watch-folder mechanism by watching to any new file that would be copied in the **00_WatchFolder_Main** folder

The script is looking for any new file every 10 seconds, and triggers the conversion for every new file copied there.

If another file is copied in the watch folder during a conversion of another file, the script will wait for the end of the conversion process and then trigger the next Conversion.

Converting a single part

- Copy and paste the CAD file into the **00_WatchFolder_Main**
- Check the result in the **10_PDF_Out** folder

Remark:

The template used in this watch folder sample is meant to present a part list document.

In case of single component conversion, the resulting PDF document may look “poor”.

Remark:

In this sample, the CAD file is moved from **00_WatchFolder_Main** into the **05_Done folder** at the end of the conversion, in order to “clean” the watch folder.

Converting an assembly

- Copy and paste all the sub-assemblies and component files into **01_WatchFolder_Components** (Note that the components of the CAD assembly located in **03_CAD_Asm_ref** are already present in that folder)
- Then copy and paste the main assembly file **_micro engine.CATProduct** from **03_CAD_Asm_ref** into **00_WatchFolder_Main**
- Check the result in the **10_PDF_Out** folder

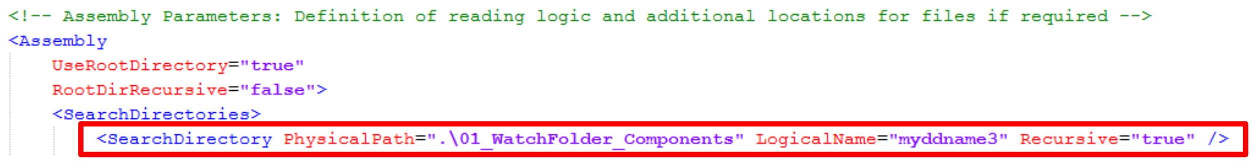
Remark:

In this sample, the CAD file is moved from **00_WatchFolder_Main** into the **05_Done folder** at the end of the conversion, in order to “clean” the watch folder.

All the component and sub-assembly files are not removed from the **01_WatchFolder_Components**

Remark:

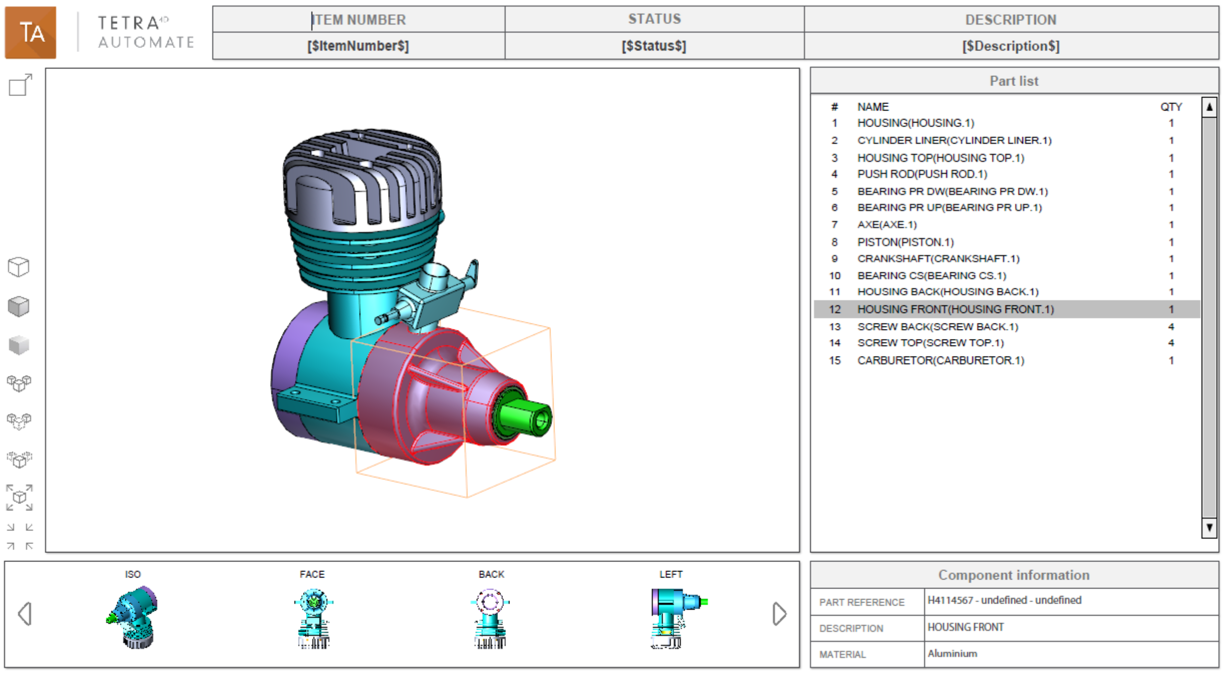
The conversion settings file has been modified to allow Tetra4D Automate find the sub-assemblies and components referenced by the main assembly
Below the additional search directory definition.



```
<!-- Assembly Parameters: Definition of reading logic and additional locations for files if required -->
<Assembly
    UseRootDirectory="true"
    RootDirRecursive="false">
    <SearchDirectories>
        <SearchDirectory PhysicalPath=".\01_WatchFolder_Components" LogicalName="myddname3" Recursive="true" />
```

Depending on the implementation and/or the organization of the CAD data (ie: components being located in several sub-folders), it could be required to define several additional search directories and to activate the Recursive Search option.

Results:



ITEM NUMBER	STATUS	DESCRIPTION
[$ItemNumber$]	[$Status$]	[$Description$]

Part list

#	NAME	QTY
1	HOUSING(HOUSING.1)	1
2	CYLINDER LINER(CYLINDER LINER.1)	1
3	HOUSING TOP(HOUSING TOP.1)	1
4	PUSH ROD(PUSH ROD.1)	1
5	BEARING PR DW(BEARING PR DW.1)	1
6	BEARING PR UP(BEARING PR UP.1)	1
7	AXE(AXE.1)	1
8	PISTON(PISTON.1)	1
9	CRANKSHAFT(CRANKSHAFT.1)	1
10	BEARING CS(BEARING CS.1)	1
11	HOUSING BACK(HOUSING BACK.1)	1
12	HOUSING FRONT(HOUSING FRONT.1)	1
13	SCREW BACK(SCREW BACK.1)	4
14	SCREW TOP(SCREW TOP.1)	4
15	CARBURETOR(CARBURETOR.1)	1

Component information

PART REFERENCE	H4114567 - undefined - undefined
DESCRIPTION	HOUSING FRONT
MATERIAL	Aluminium
