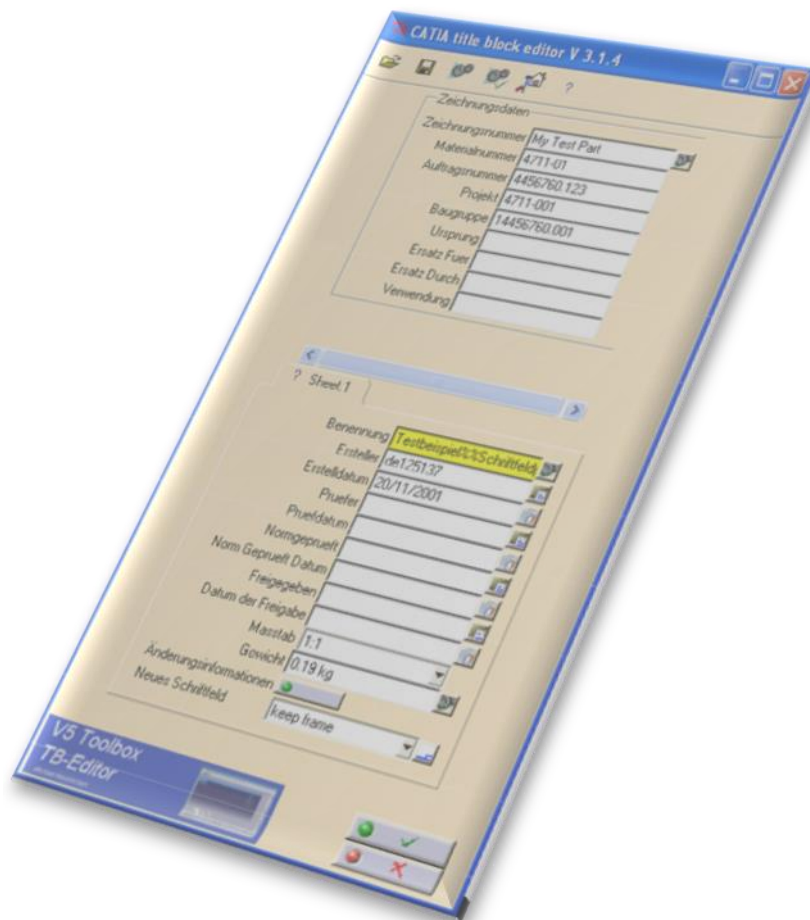




CATIA V5 Title Block Editor V 4.4.8


















Michael Neukirchen


mailto :michael.neukirchen@3ds.com




Content

1	Preface	6
1.1	Acknowledgments	6
1.2	What's new	7
2	Part I: User Guide	12
2.1	First start	13
2.2	Call the Macro	14
2.3	First Dialogues	15
2.3.1	 Diagrams	15
2.3.2	Empty Background View	16
2.3.3	Multiple Configurations	16
2.4	Edit Title Block	17
2.4.1	Copy & Paste info between fields	18
2.4.2	Insert special characters	18
2.4.3	The different options on the main panel	18
2.4.3.1	Buttons on the top of the Panel	18
2.4.3.1.1	 Open File	18
2.4.3.1.2	 Save File	18
2.4.3.1.3	 Show 3D models linked to the drawing	18
2.4.3.1.4	 Accept all 3D infos (optional)	19
2.4.3.1.5	 TD1 compatibility	19
2.4.3.1.6	 Configuration (optiona)	20
2.4.3.1.6.1	BackGround View	20
2.4.3.1.6.2	Units	20
2.4.3.1.7	 Help	21
2.4.3.2	 Copy data from	21
2.4.3.3	Add revision info	21
2.4.3.4	 Default Values	21
2.4.3.5	 Default Values from the 3d master	22
2.4.3.6	 Date Chooser	24
2.4.3.7	Mandatory Fields	25
2.4.3.7.1	Enabled	26
2.4.3.7.2	Forced	26
2.4.3.8	 New Drawing Frame	26
2.4.3.8.1	Selection of a frame icon	27
2.4.3.8.2	Selection of the drop down list	27
2.4.3.8.3	Selection of the ok button	27
2.4.3.8.4	Selection of the cancel button	27
2.4.3.9	Modify Bom	27
2.4.3.10	Handling of CATIA tables	28
2.4.4	Sample Table	28
2.5	Missing fields	30
2.5.1	Change	30
2.5.2	Cancel	31
2.6	 Use Cases	31
2.6.1	Empty Model	31
2.6.2	Model containing a frame	32
2.6.3	Model with something else in the background view	33
3	Part II: Admin Guide	34
3.1	 Installation	35
3.1.1	General Client / Server installation	35



3.1.2	National Language Support	35
3.1.3	Windows / Unix Change of the TitleBlock.Catscript file	35
3.1.4	Unix	36
3.1.5	Ready to Go!	36
3.2	How does it work?	36
3.2.1	In general	36
3.2.2	CATIA documents supported	37
3.2.3	SMARTEAM compatibility	37
3.2.4	Support of Library Components	38
3.2.5	Multiple configurations	38
3.3	 Customization	39
3.3.1	Unload directory	39
3.3.2	English Panels	40
3.3.3	Add a new drawing frame	40
3.3.4	Overview	41
3.3.5	Configuration data sets	44
3.3.5.1	Basic configuration – basics.cfg	44
3.3.5.2	title_block_config.tcl configuration data set	44
3.3.5.2.1	Sample 1: small config	45
3.3.5.2.2	Sample 2: complex config	46
3.3.5.3	Fields	50
3.3.5.4	Global fields	50
3.3.5.5	Extended field definition	50
3.3.5.5.1	attribute entry_width, entry_alignment	50
3.3.5.5.2	attribute Mask	50
3.3.5.5.3	attribute Help	51
3.3.5.5.4	attribute Mode	51
3.3.5.5.5	attribute pass3dProperties	51
3.3.5.5.6	attribute Default	52
3.3.5.6	Revisions	52
3.3.5.6.1	Configure the revision panel	52
3.3.5.7	Units	52
3.3.5.8	Width of entry fields	53
3.3.5.9	Definition of the height/width of the notebook area	53
3.3.5.10	Definition of the height/width of the frame chooser	53
3.3.5.11	Date Format	54
3.3.5.12	Areas	55
3.3.5.13	Special Character	55
3.3.5.14	Special Variables Containing CATIA infos	55
3.3.5.14.1	CATIAVERSION	55
3.3.5.15	Pre-selection of the Modify Bom entry	55
3.3.5.15.1	Number of Sheets	56
3.3.5.15.2	Number of Actual Sheet	56
3.3.5.15.3	Scale of the sheet	56
3.3.5.15.4	Values from the 3D-Master	57
3.3.5.15.5	Summary of pre configured values	58
3.3.5.16	Automatisms for TB entries	59
3.3.5.17	TD1 compatibility – td1_config.cfg	60
3.3.5.18	Changing Text Names	61
3.3.5.19	Modifying the GUI – skin.tcl	62
3.3.5.20	NLS Support	62
3.3.6	BOM management	62
3.3.6.1	Get a Bom table	63
3.3.6.2	Configure your Bom	64
3.3.6.2.1	Tag – Component Id's first item	65
3.3.6.2.2	Type	65
3.3.6.2.3	Head	65
3.3.6.2.4	Col	65
3.3.6.2.5	How to support Units within the BOM – Weight, WetArea, Volume	67
3.3.6.2.6	How to support user entered data	67
3.3.6.3	Override OneLevel / AllLevels	68



3.3.6.4	What to do if something doesn't work.....	68
3.3.6.5	Known Issues.....	68
3.3.7	Temporary data sets.....	68
3.3.8	Sample: Add a new entry.....	69
3.4	Batch exchange of title blocks.....	69
3.5	Upgrades.....	70
3.5.1	Upgrade to version 4.....	70
3.6	 Error, Enhancement reporting.....	70
4	Samples.....	71
5	Part III: Appendix.....	72
5.1	How to use a button to call a CATIA macro.....	72
5.2	Known issues.....	74
5.2.1	PaperSize mismatch.....	74
5.2.2	Geometry in the background.....	74
5.3	Hints and Tips.....	75
5.3.1	How to use several Notes with one title block entry.....	75
5.3.2	Using the CATIA DS title block macro environment.....	75
5.3.2.1	The macro hangs somehow or does nothing.....	75
5.3.2.2	The published methods are not selectable.....	75
5.3.2.3	I don't see any macros within the macro environment.....	75
5.4	Copyright Notice, Terms and Conditions.....	76
5.4.1	GNU GENERAL PUBLIC LICENSE.....	76
5.4.2	Preamble.....	76
5.4.3	TERMS AND CONDITIONS.....	77
5.4.3.1	0. Definitions.....	77
5.4.3.2	1. Source Code.....	78
5.4.3.3	2. Basic Permissions.....	78
5.4.3.4	3. Protecting Users' Legal Rights From Anti-Circumvention Law.....	79
5.4.3.5	4. Conveying Verbatim Copies.....	79
5.4.3.6	5. Conveying Modified Source Versions.....	79
5.4.3.7	6. Conveying Non-Source Forms.....	80
5.4.3.8	7. Additional Terms.....	81
5.4.3.9	8. Termination.....	82
5.4.3.10	9. Acceptance Not Required for Having Copies.....	82
5.4.3.11	10. Automatic Licensing of Downstream Recipients.....	82
5.4.3.12	11. Patents.....	83
5.4.3.13	12. No Surrender of Others' Freedom.....	84
5.4.3.14	13. Use with the GNU Affero General Public License.....	84
5.4.3.15	14. Revised Versions of this License.....	84
5.4.3.16	15. Disclaimer of Warranty.....	85
5.4.3.17	16. Limitation of Liability.....	85
5.4.3.18	17. Interpretation of Sections 15 and 16.....	85
5.5	Sample Product.....	86
5.6	ASCII Table.....	86
5.7	TCL String command.....	87
6	Index.....	92

Pictures

Picture 1:	Sample drawing frame.....	13
Picture 2:	Call a macro in CATIA step 2.....	14
Picture 3:	Diagram Mode.....	15
Picture 4:	Options to allow Diagrams.....	15
Picture 5:	Frame selector.....	16
Picture 6:	Multiple configs.....	16
Picture 7:	The main panel.....	17
Picture 8:	Special Character window.....	18
Picture 9:	Link to 3D models.....	19
Picture 10:	TD1 configuration.....	19



Picture 11: Missing parameters	20
Picture 12: Disconnected parameters.....	20
Picture 13: Select Units	20
Picture 14: Adding change infos.....	21
Picture 15 3D info differs from actual title block entry	22
Picture 16 Checkbox to retrieve data from 3d	22
Picture 17: Unit added to weight	23
Picture 18: 3D Value differs	24
Picture 19: 3D Value and Value in the Drawing are equal	24
Picture 20: Date chooser.....	25
Picture 21 TB with mandatory Fields	26
Picture 22: Change of drawing frames	26
Picture 23 Selection Panel for Frames.....	27
Picture 24 Modify Bom Checkbutton.....	28
Picture 25 Bom Info message	28
Picture 26 Table in CATIA Drawing.....	29
Picture 27 Button to edit the table	29
Picture 28 Table Editor.....	30
Picture 29: Missing Elements in the drawing frame.....	30
Picture 30 Unload structure	40
Picture 31: Selection of drawing frames.....	41
Picture 32: CATIA component names.....	42
Picture 33: Option Setting to get the Weight Info	43
Picture 34 Entry panel with CATIA version and release.....	56
Picture 35 User defined property	57
Picture 36 Value differs 3D and actual entry	58
Picture 37 Component Id for Bom Table.....	64
Picture 38 CATIA Bom Headers	66
Picture 39 CATIA Bom Properties.....	66
Picture 40: Sample for Bom with Weight info	67
Picture 41 Sample title block.....	71
Picture 42 Toolbar definition.....	72
Picture 43 Icon definition	73
Picture 44 Paper Size Mismatch	74

1 Preface

The title block editor provides an example (it was small in the beginning) for an easy integration of CATIA V5 with other applications. The program is free and can be distributed and used free of charge. The copyright follows the GNU copyright.

The program consists of 2 parts:

- 1) a CATScript macro which extracts the title block data from the CATIA drawing and which performs the modifications
- 2) a TCL/TK program which displays the GUI and allows an easy modification of the data.

Both programs run under Windows (currently tested on Windows XP and in intervals on Unix). The script is wrapped into an executable .exe file (for the Unix platforms you can download the executable as well), so the installation effort is minimal. Any help/experience is welcome.

1.1 Acknowledgments

Thank's a lot to

- + Martin Röcker (thanx for the parameter tips and the Diagram support)
- + Stephane Reautey (thanx for the UNIX testing and doc!!)
- + Daniel Schnorf (ideas for correct termination of the GUI, Performance issues)
- + Jorge Mallea Catala
- + John B. Cearns for the thorough testing of multiple sheets, 3.2.16 and the contributions to the bom management and unit support
- + Thomas Pfaff for the testing of the CADSystem variable
- + G. Scheffler for the sheet numbering and extraction of properties
- + J. Völkel for the input to mandatory fields
- + J. Völkel, D. Reck, J.B. Cearns, P. Varadharajan, S. Gellet for the test of the 4.0.0 version
- + Alexandre le Gac for the support on tables
- + Varadharajan, Prabhakaran for his thorough testing and valuable feedbacks on BOM handling and scenes
- + H. Könen for the support on the catiav5forum
- + Boris Ackermann for the ideas and support on callbacks.
- + Guillermo Alvarez for supporting the bug fixes in 4.4.5

For testing, suggesting and reporting.

I would be happy to get some sample screenshot for the usage of the macro. These samples will be documented within this documentation and help the users to design their configurations.



1.2 What's new

Summary of the history








Release	Remarks
2.2.0	Add a title block based on a template Exchange a title block Scrollable Notebook area for large title blocks
2.3.0	Specification for fields which are valid and identical for all sheets (drawing level fields) Check of availability of all title block fields additional fields referring to 3d data description for V5R8 added
2.3.1	Error correction for the exchange of frames Display of the list of 3d models for generative drawings
2.3.2	Allowed to change all revision entries. New revisions will be added on top of the old ones. Macro does not stay in the Background view any more
2.3.3,2.3.4,2.4.0	Lots of changes for Unix compatibility New date chooser to avoid iWidget set, meaning standard tk + Bwidget set is all you need. Possibility to use CATIA parameters linked to text. This is the way TD1 exchanges the data.
2.4.1	Improved parameter handling (performance) Check and information for inconsistencies regarding parameters Check for termination by selecting the Windows "x" Switch from foreground mode to background mode → CATIA repaints its window.
2.4.3	GUI configurable through skin.tcl Background/Foreground through basics.cfg UNIX compatibility
2.4.4	Editor works, even if no revisions are defined (all relevant elements are hidden) Debugging
2.4.5	Bug Fix
2.4.6	Doc Changes Bug fix for revision changes Support for scrolling on the revision panel (the size of the panel can be configured in the title_block_config.tcl data set)
2.4.7	In case of missing fields a choice for a new frame is offered. Tracing capability added NLS support for German and English (any translation is welcome and will be added) Ok, cancel buttons replaced by graphical buttons (avoids language dependant buttons)
2.4.8	Scrolling of Drawing Data Area enabled



Release	Remarks
2.4.9	Default values for revisions enabled Compatibility to V5R11 – 2.4.8 causes problems
2.4.10	Bug Fix
2.4.11	Support of Starkit on Windows, resulting in a smaller package Support of dedicated location name for the CATIA template drawings
2.4.12	Bug fix for multiple sheet drawings, ANSI/ISO issues Support of components from libraries
2.4.14	Bug fixes Possibility to include CATIA Version
2.4.15	Bug fixes
2.4.16	CATIA infos added
2.4.17	Documentation change (error fro SheetNum → NumSheets) Entry width for panels configurable
2.4.18	Documentation change (Tip added from John) Bug fixes (NumSheets, Scale > 1)
2.5.0	Prompt to retrieve 3D analysis data (mass etc.) to avoid long response times Retrieval of User Properties from the 3D master.
2.5.1	Access to 3D info for each sheet individually Bug fixes
3.0.0	Bom Management included (see 3.3.6)
3.0.1, 3.0.2	Bug fixes, 3.0.2 circumvents a bug in the API (introduced ~ V5R12SP9). Tested on R14 level.
3.0.3	Translation supported for messages coming from the CATScript part of the macro. Bug fixes Doc part of customization restructured Configuration for the prefix of Tblock entries added.
3.1.0	Unix Support with Starkits
3.1.1	Bug fixes, Check for active document is drawing Support of Description field from 3D
3.1.2, 3.1.3	3.1.2 first code parts for a general table editor (streaming) Bug fix for BOMs, which have the title line at the top, Direction of BOM top down
3.1.4	First version for a table editor added Link to 3D in the case when the user selects a partbody first corrected. Support for Units added
3.1.5	Link to 3D in the case when the user selects a partbody first corrected – now it works.
3.2.0	Support of Diagrams, support for R16. Any version below 3.2.0 will not work on R16 due to a naming conflict (FromSelection)
3.2.1	Bug fixed for ST parameters
3.3.0	Bug fix for ST parameter mapping Introduction of mandatory/optional fields



Release	Remarks
4.0.0	<p>Multiple configurations, rework of config directories</p> <p>Scenes check: Drawing derived from scenes</p> <p>Preserve of User entered data in BOM</p> <p>Table Editing: columnWidths automatically adjusted to maximum entry</p> <p>If no document is active, a blank CATDrawing will be created automatically</p> <p>Doc: Within the index you will find the current updates under the entry "update 4.0.0"</p> <p>The TitleBlock GUI will stay on top of all applications.</p> <p>Bug fixed for ST parameter mapping</p> <p>Bug fixed for details</p>
4.0.1	<p>Bug fixing for revisions</p> <p>Table handling enhanced (insert / delete rows)</p> <p>Revision Block:: Cancel Option added</p>
4.0.2	<p>Cut, Copy Paste like windows (Control c,v,x)</p> <p>Bom Table hides in those cases of a generative sheet of a CATPart or no generative sheet. This enables the usage of one template for Parts, Products and interactive 2D designs. Ability to show/hide a BOM Table.</p> <p>Extended Field definition: entry_mode added to give the possibility to disable fields (so the user cannot edit the field)</p> <p>NoShow possibility for the buttonBox at the top of the TB</p> <p>Error reading files solved.</p>
4.0.3	<p>Support of text fields for multiple line entries</p> <p>Check, if a l_variable field is defined in case of list fields.</p> <p>Display of PartNumber in BOM wrong (HTML code)</p> <p>Open error in line 788</p> <p>TB editor does not work in 64 bit environment (windows)</p> <p>Mode forced does not work in a new drawing</p> <p>Mass problem for german environment solved</p> <p>TB should offer the possibility to display 3D document name (incl Path) from cad.de</p> <p>➔ TitleBlock_Text_3DmodelFullName, TitleBlock_Text_3DmodelName added</p>
4.0.4	<p>Special character window added e.g. to enter diameter signs</p> <p>Change Panel updated</p> <p>Documentation restructured in a User Part and an Admin Part</p> <p>Support for CADDWGModelName</p> <p>Option, to stream all Properties from 3D to the editor without UP prefix.</p> <p>Extended field definition to directly insert the values from 3D into the GUI (without the need to press the button)</p> <p>Problem with mass (". " And ", " problem) fixed.</p> <p>Material property supported for generative Drawings</p>
4.1.0	<p>Bug fixes for material and array problem for multiple sheets</p> <p>Bug fix for large number of TB Entries</p>

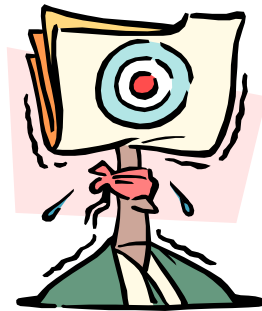
Release	Remarks
	<p>Bug fix for special CharacterWindow within the Change Panel</p> <p>Bug fix for scenes</p> <p>Bug fix for AllLevel Boms</p> <p>Modify Bom pre-selection configurable</p> <p>User selection enabled to switch between OneLevel and AllLevels for BOMs</p> <p>Width etc. configurable for the frame selection panel</p> <p>The frame selection panel will show a sorted list of frames</p> <p>Associated 3D model will be automatically be loaded -> that fixes the problem with mass calculation (configurable in Basics.cfg)</p> <p>CADDWGModelNameWithoutExtension supported, which returns the Drawing name without extension</p> <p>Mapping possibility added, to map 'legacy' fields to TB-field names</p> <p>Delete for Revision entries optional</p> <p>Cut/copy/paste behavior for Windows enhanced (Shift Control works too)</p> <p>Tables work on AIX</p>
4.1.1	<p>Bug fixes</p> <p>User selection for BOM OneLevel, AllLevels reworked</p>
4.1.2	<p>Bug fix: path names for windows did not show the "\"</p> <p>Bug fix: ProductScenes do work now</p> <p>Bom: Selection of no Sow Bom will deselect the other Bom related switches</p> <p>Call Back added for TB-entries: Automatisms possible</p>
4.1.3	Bug fixes
4.2.0	<p>Library restructuring (internal)</p> <p>Configuration option added to avoid leading zeroes for sheet numbers</p> <p>Problem with Unicode fixed ('TBConfig error')</p>
4.2.1	<p>Bug Fixes</p> <ul style="list-style-type: none">  Error TBeExit for the change panel  Error leadingZeroes  Show the right config within the change panel for multiple configs.  CATIA_TEMPLATES <p>Added the option Default Values for extended field definition.</p>
4.3.0	<p>Bug Fixes</p> <ul style="list-style-type: none">  Drawings derived from CATParts didn't show the relation to the 3D model <p>Sorting for AllLevel Bom enabled</p>
4.3.1	<p>New mode 'hidden' for TB fields: Setting the Mode to hidden, will skip the display on the panel. This is useful, if you want to import data from the 3D model and process the data using the callback mechanism and fill in other fields depending on derived values.</p>
4.3.2	<p>Bug Fixes</p> <ul style="list-style-type: none">  Error for linked 3D models in VPM context  Error field length for field being not mandatory
4.3.3	Bug Fixes

Release	Remarks
	<ul style="list-style-type: none"> ✚ Error for piping / tubing R18 (Thanks to Gmür Philipp from Helbling!) ✚ extFieldDef – “Default” problem fixed
4.4.0	<p>English sample for a config data set provided, manual changed accordingly</p> <p>IBMTitleblock.CATScript renamed to Titleblock.CATScript</p> <p>Internal renamings.</p>
4.4.1	Mapping of texts did only work for one sheet
4.4.2	The NumSheets variable was always set to 1. Bug fixed
4.4.3	Sporadic error in translateinit fixed
4.4.5	Bug fix for tables
4.4.6	Removed the constraint for installation (allowing blanks in directory name)
4.4.8	Bug fix for default customization



2 Part I: User Guide

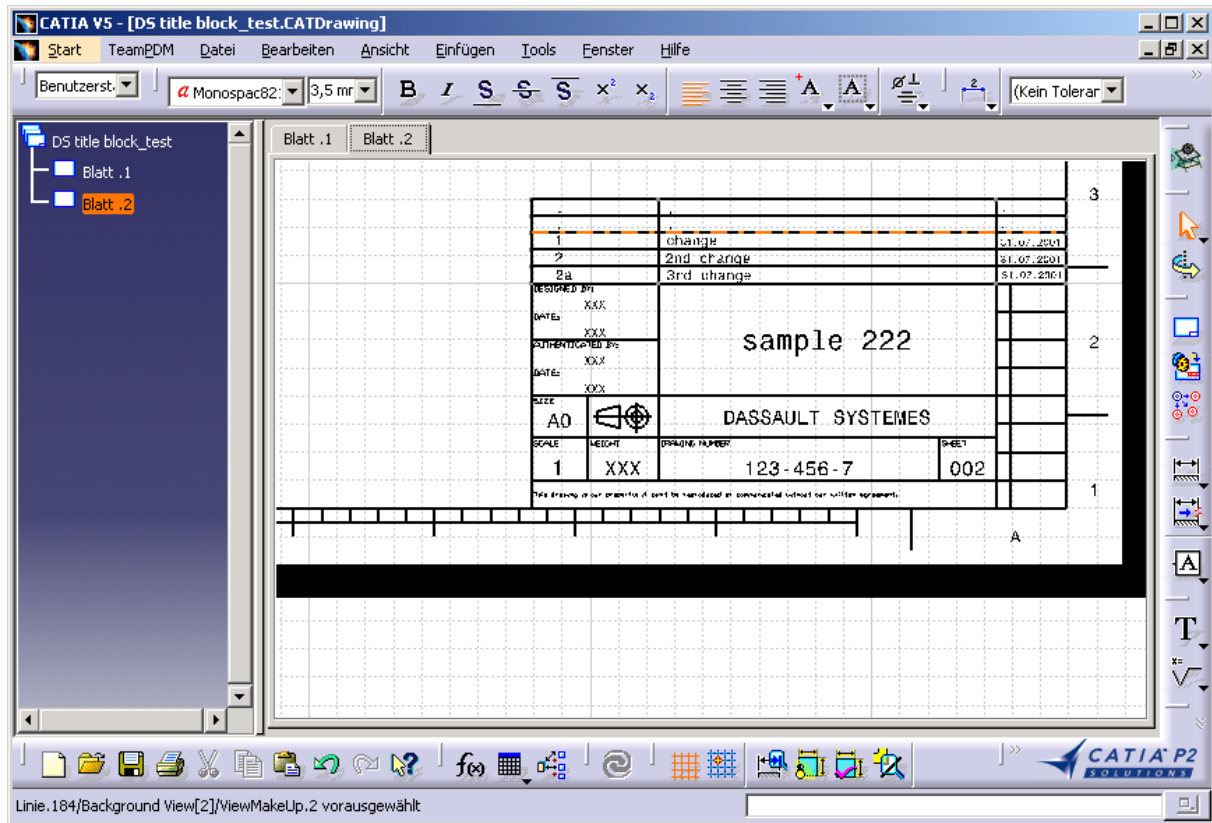
CATIA V5 Title Block Editor V 4.4.8 User Guide





2.1 First start

Start a new drawing and run the macro from Tools/Macro. The result, after filling the title block entries will be similar to:



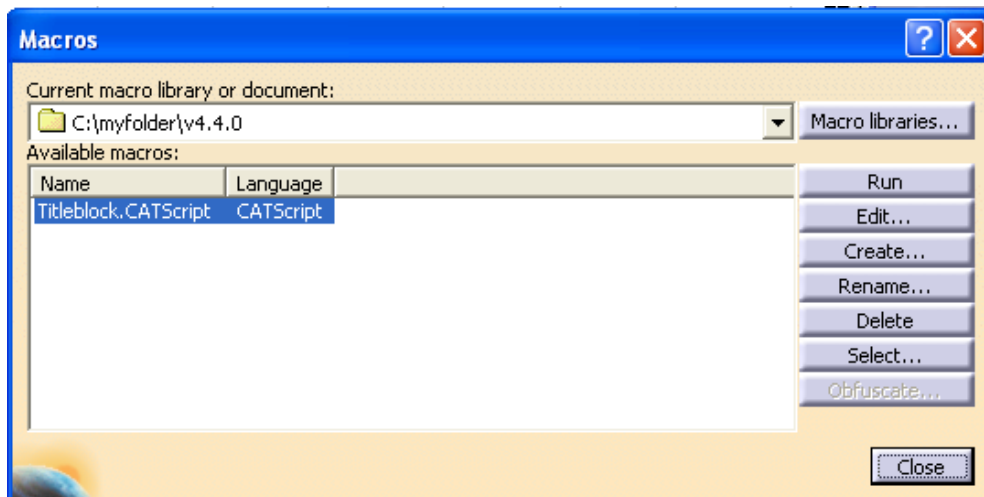
Picture 1: Sample drawing frame

The following chapters will guide you through the process in detail.

2.2 Call the Macro

You can start the macro either by calling it directly or by using a dedicated button in the command area of CATIA (see chapter 8).

Select the directory, where you have installed the title block macro (Library). Then you will see the Titleblock.CATScript macro.



Picture 2: Call a macro in CATIA step 2



2.3 First Dialogues

After starting the TitleBlock program, it depends on the kind of model you have active.

- Diagrams
- Drawings with an empty background
- Drawings, which already contain a 'proper' title-block

If you have decided to run with multiple configurations, and you have an active drawing containing TB data, the macro will check which configuration fits best to the drawing.

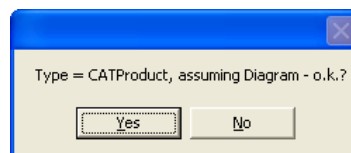
If no document is available, i.e. you have a session with no CATIA document, the program will create a blank CATDrawing.

2.3.1 Diagrams

Diagrams (Tubing, P&ID, HVAC, ...) are stored within CATProduct type files. If you call the macro and a CATProduct is active, the macro will assume that you are working on diagrams.

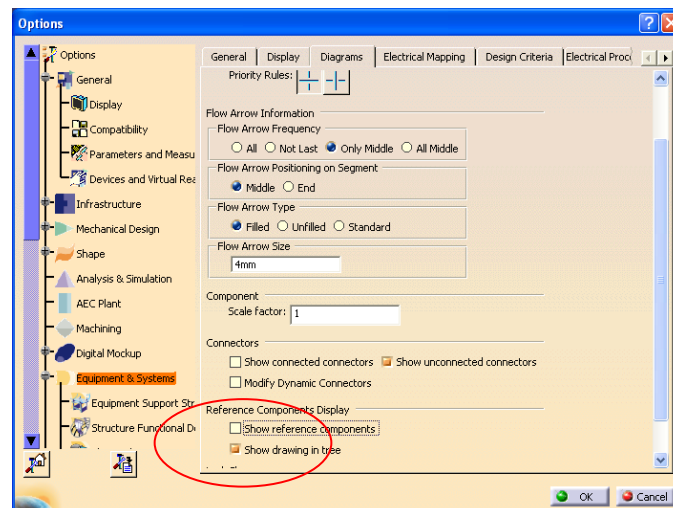


Just to be sure, the macro will pop up a window to confirm the diagram mode.



Picture 3: Diagram Mode

The macro needs access to the drawing area. So you have to activate within the tools options the toggle "Show Drawing in Tree".

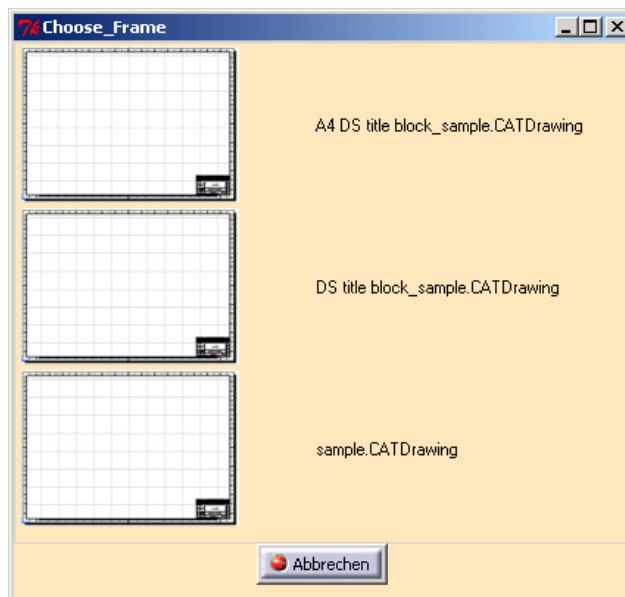


Picture 4: Options to allow Diagrams

Bad News: In CATIA V5R15 the macro will only work on diagrams starting with SP4. On R16 you will need SP1 at least. Some bugs in lower levels of the CATIA software don't allow the usage of the TB-Macro for Diagrams.

2.3.2 Empty Background View

If you have an active sheet with an empty background view, the macro prompts you to select a drawing frame (may be slightly different due to customisation). This will be the case, if you start a new drawing or insert a new sheet into an existing drawing.



Picture 5: Frame selector

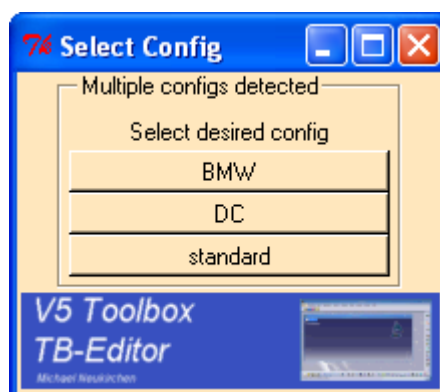
The left side shows a small image of the Title Block, on the right side you will find the CATIA names of the drawings.

You can select one of the images or the name. A selection of the Button will cancel the method. If you have more frames, the panel will have scrollbars.

2.3.3 Multiple Configurations

In case you have defined multiple configurations “,, **and** the current frame content fits to more than one configuration, the TB macro needs your help. This can be the case, when you have to deliver your drawings to different OEMs such as DC, BMW,

The TB will display a list of configs, which match to the content of your current active drawing.

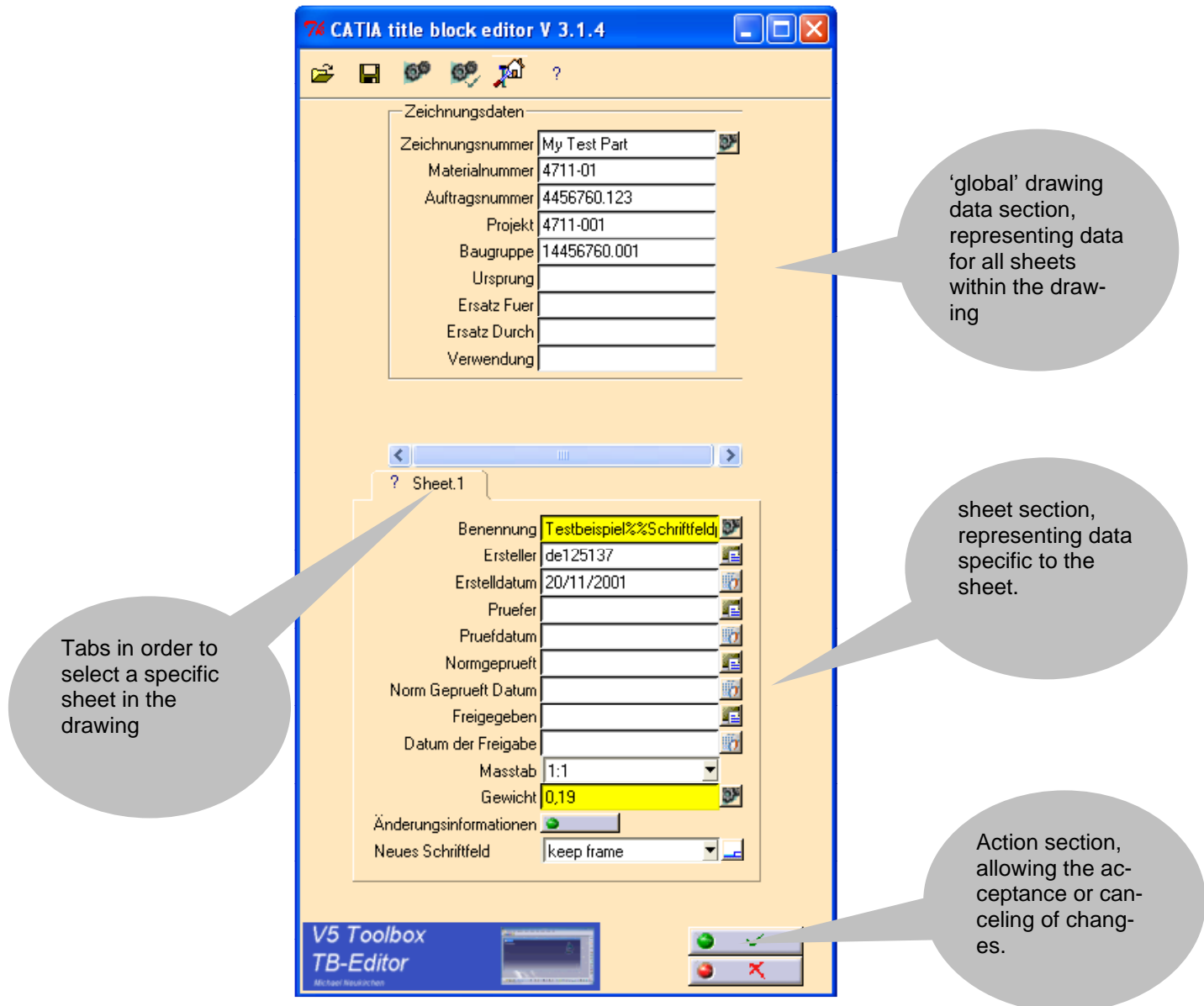


Picture 6: Multiple configs

2.4 Edit Title Block

The Macro scans the model and shows the GUI to enter / change data. If you select a pushbutton near to an entry field, a default will be put into the entry area.

All descriptions -labels- are configured in the config data set, to ensure a maximum of flexibility (see the admin section).



Picture 7: The main panel

2.4.1 Copy & Paste info between fields

You can use the standard behavior to copy and paste data between different entry fields. On Windows you can use Control-c, Control-v, Control-x.

On top you can use a drag and drop mechanism by selecting the text you want to move to another entry field and drag that text to a target field (keep mouse button 1 pressed and drag).

2.4.2 Insert special characters

Not all characters can be entered via the keyboard. You can use the Alt key together with the s key to show a panel with special characters (can be customized by your admin).



Picture 8: Special Character window

Select one of the symbols shown and the symbol will be inserted at the current cursor position in the current entry field of the TB editor.

The cancel button will close the panel without inserting a character.

2.4.3 The different options on the main panel

The main panel will display a user interface to allow the modification of the title block fields.

2.4.3.1 Buttons on the top of the Panel

The buttons at the top of the panel allow general actions for the TB environment such as

- ✚ Save and load of TB data
- ✚ Configuration Options
- ✚ General acceptance of 3D data
- ✚ General Info on ST interop
- ✚ Help

This button bar can be enabled or disabled.

2.4.3.1.1 Open File

This button allows the import TitleBlock entries of a previous stored sheet into the current sheet.

2.4.3.1.2 Save File

 saves the contents of the current sheet's TitleBlock entries to disk. A save as dialog appears to select the file. You can use this option, to import data into a blank model. (Previously you have to export data with the save option).

With Open/Save you can perform a copy/paste style of action for TB-entries from one Drawing to another.

2.4.3.1.3 Show 3D models linked to the drawing

Here you can display a list of the 3D models which are linked to the drawing



Picture 9: Link to 3D models

The sample shows the original link for the drawing on my machine.

2.4.3.1.4 Accept all 3D infos (optional)

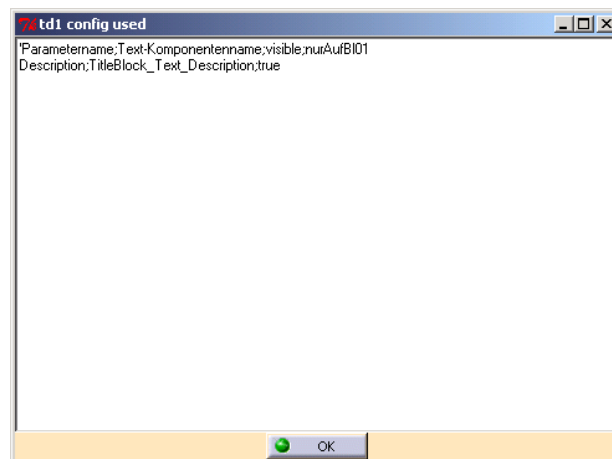
You can accept all data which comes from the 3d associated model in one shot. All 'yellow' entry fields will be filled with the info from 3D.

Yellow fields show discrepancies between entries in the TitleBlock fields and the data in the corresponding 3D model.

This menu item is optional – depending on the configuration.

2.4.3.1.5 TD1 compatibility

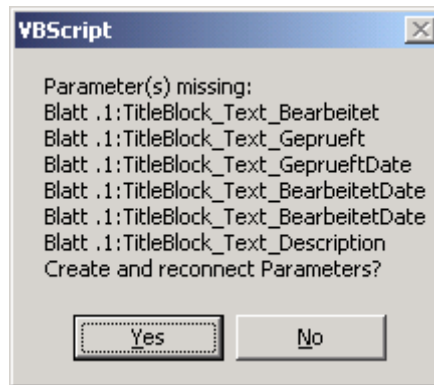
As soon as the TD1 configuration data set is available in the config directory, the TD1 icon is shown. If you select this icon, you will see the configuration data set.



Picture 10: TD1 configuration

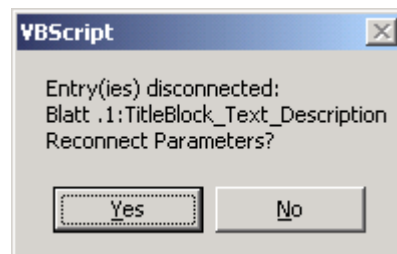
The macro checks the parameters defined in the config data set against the actual values. It detects the following problems

- Missing parameters. This can happen, if a user has deleted a parameter, or additional parameters have been defined for a TD1 integration.



Picture 11: Missing parameters

- Parameters which are not longer connected to a text entry (i.e. parameter value is different from text entry). The disconnecting of parameters can happen, if a user edits a text field directly instead of modifying the corresponding parameter.



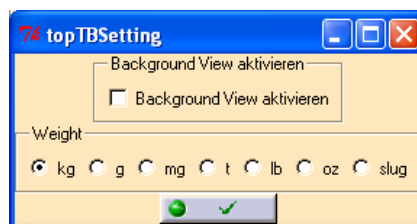
Picture 12: Disconnected parameters

In both cases, the user can accept corrective actions. Then the macro will delete the defined parameters, recreate them and connect the parameters to texts. Depending on the total number of parameters in the drawing (e.g. lots of views), this will take time.



2.4.3.1.6 Configuration (optional)

If you allow the user to select this menu item, he will get the following panel:



Picture 13: Select Units

2.4.3.1.6.1 *BackGround View*

A selection of the Background View activate toggle will leave the background active, when you leave the macro. This is important in the case of Diagrams – it is the only way to access the background, if you want to change something manually.

2.4.3.1.6.2 *Units*

You can specify another unit within the unit config dialogue. The units shown are dependant of the configuration.

After you select a unit, the TB macro will ask you, if all values from 3D should be accepted. If you confirm here, all values will be accepted (the yellow field will vanish) and the weight, volume and wetarea fields are re-calculated.




2.4.3.1.7 ? Help

Shows this doc in pdf format.

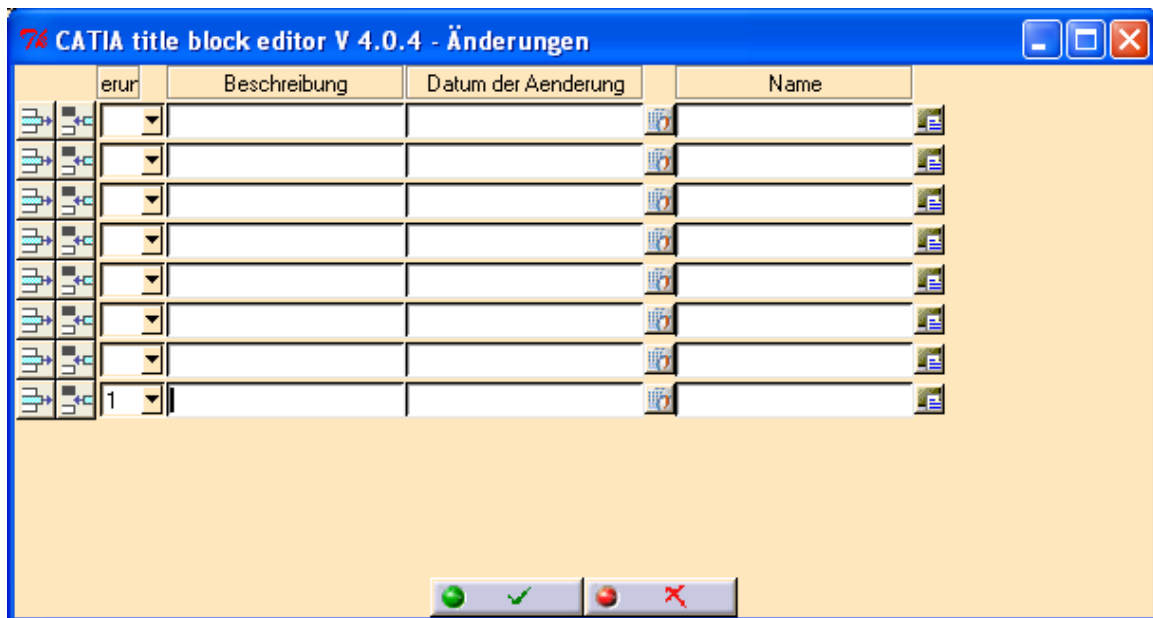
2.4.3.2 Copy data from

If the CATIA model contains more than one sheet, the macro offers the possibility to copy the contents from another sheet. This can be used in cases of new – blank – sheets to quickly fill most of the fields.



First select the arrow to choose the desired sheet. This sheet will be the source of the data. After pressing the copy button  the values will be copied to the actual page.

2.4.3.3 Add revision info

The makro shows the change history, as far as it is stored in the model (the cat_title_block_change fields). The entry fields allow you to enter / change data for the revisions.




Picture 14: Adding change infos


On the left hand side, you can select for each line a button to add a line  or delete a line .

The cancel button will reset the values and close the panel. The ok button will accept the entries done.

2.4.3.4 Default Values

A selection of the default Button  will enter the default value into the corresponding entry field. Default values can be specified in the configuration data set (see below).

2.4.3.5 Default Values from the 3d master

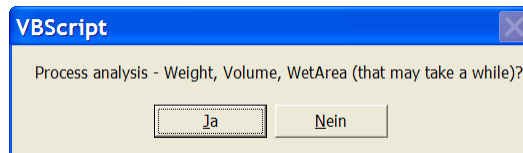
If data from a 3d master is available, i.e. at least one view in the sheet is a derived view, the editor offers the data by the  icon. The entry field will be yellow in those cases, the 3d info differs from the actual title block entry:



Picture 15 3D info differs from actual title block entry

Pointing with the mouse on the Button will display the 3D info. Clicking the button will copy the 3d info data into the entry field. The editor takes the **last** derivative view found in the sheet as the source for this info.

In case the drawing is derived from a CATProduct, the macro will prompt the user to decide whether 3D analysis data such as Weight, Volume and WetArea should be retrieved from the 3D master. This can take a while, since CATIA will recalculate these items.



Picture 16 Checkbox to retrieve data from 3d

Since 3.1.4 you can convert the CATIA units, which are given in the mks-system to other units. These units can be specified in the configuration dataset.

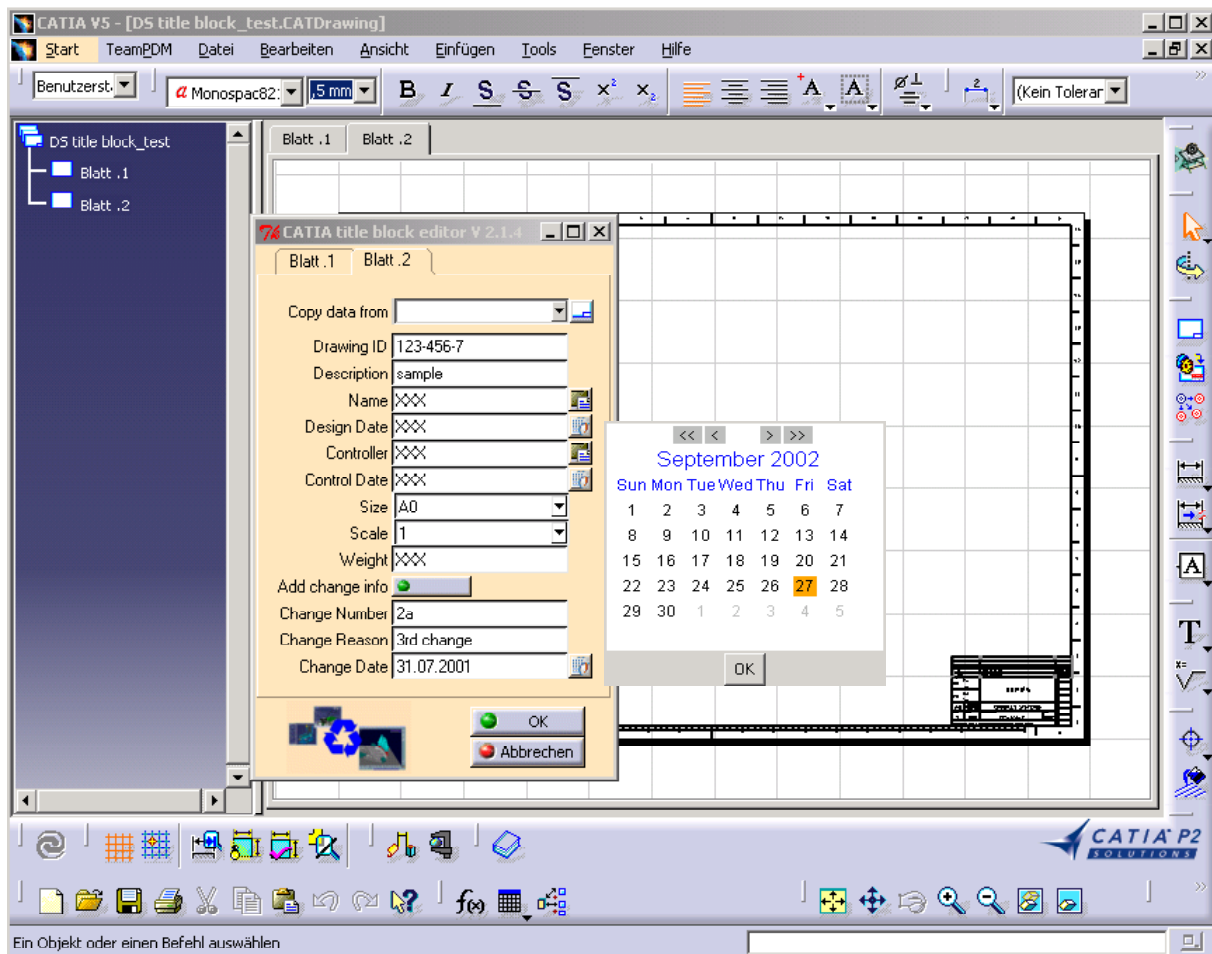
Picture 17: Unit added to weight

You can switch the unit, by selecting the appropriate values within the configuration dialogue ().

Automatic insertion of properties

Your Admin can decide that specific values will be directly inserted into the TB-Editor. Colors will signal, that the data is originated from the 3D model (green color, if data from 3D and drawing from the drawing are equal, red color, if the values differ). A sample below is provided.





Picture 20: Date chooser

In order to select the current date, select it and then the ok Button

2.4.3.7 Mandatory Fields

Beginning with 3.3.0 the administrator can define mandatory fields. These fields will be marked by red or green dots.

- signals, that the field has a value
- signals, that the field is empty and a value is expected from the user

Depending on the mode the administrator has chosen, the macro will prompt the user for the missing values.

Since CATIA doesn't allow real empty fields for texts, a so called "empty char" has been introduced. Usually this character will equal to "-". Each field, which has only this "empty char" will be treated as empty and the user will get a hint or will be forced to enter data for this field.

A field can be defined as mandatory within the extended field definition.

Sample:

CATIA Title Block Editor 448.doc



Picture 21 TB with mandatory Fields

2.4.3.7.1 Enabled

In case the macro is running in the “enabled ” mode, the macro will issue a warning, that values are missing. The user can decide to continue anyway – leaving some mandatory fields empty.

2.4.3.7.2 Forced

In that case the user cannot leave the macro by the ok button until all mandatory field are filled in. The only way to leave the macro is to select the cancel button, discarding all changes which may have been done.

2.4.3.8 New Drawing Frame

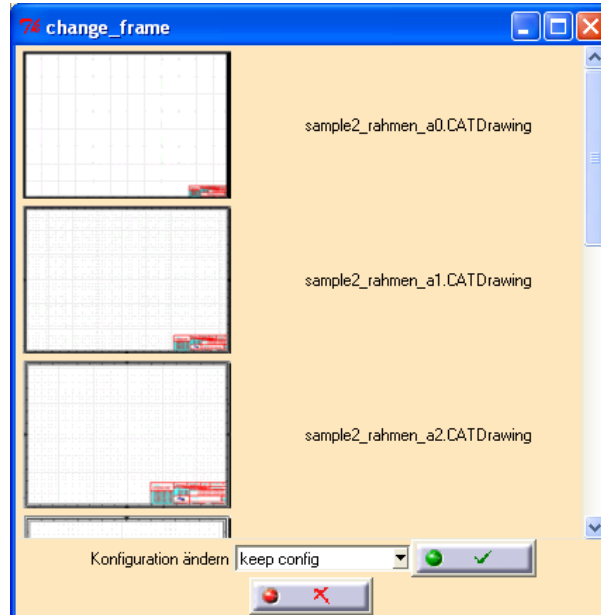
You can select a new drawing frame for the active sheet (you won’t find this option on the tabs of the other sheets). In case of a selected button, the macro will replace the drawing frame of the active sheet by a new one. All entries which are in the edit environment will be put back onto the drawing frame.

Picture 22: Change of drawing frames

The box shows the possible frames, if you select the button on the right the panel for the selection of the frames will appear. The “keep frame” title block will preserve the current drawing frame.

If you want to work this way, entries such as A4 should not be editable but a fix component of the frames.

If you select the little frame icon, the selection box for the selection of frames will pop up.



Picture 23 Selection Panel for Frames

2.4.3.8.1 Selection of a frame icon

If you select one of the frame icons, the current frame in your drawing will be exchanged. Usually you can do that to increase/decrease the size of a frame.

2.4.3.8.2 Selection of the drop down list

The drop down list offers to change the configuration for the TB macro. This drop down is only shown, if you are using more than one configuration. This targets at the scenario that you have to deploy your drawings to several OEMs.

After selection of a different configuration, the displayed Icons will change and you will be able to select a drawing frame.

2.4.3.8.3 Selection of the ok button

In case you start the TB macro with a configuration that does not fit to your current drawing, you will get a lot of errors (see missing fields below). If you only want to change the configuration and not exchange the drawing frame, then select the desired configuration and press the “ok” button.

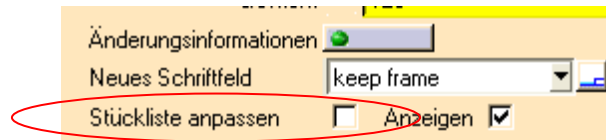
2.4.3.8.4 Selection of the cancel button

The panel will disappear without changes.

2.4.3.9 Modify Bom

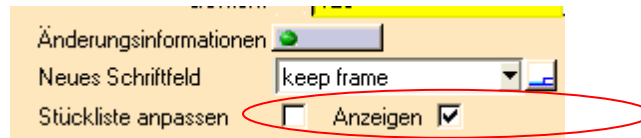
If your drawing has a table with a component ID: MyBomTable, then the macro offers you the possibility to rebuild the bom automatically from the 3D master product.

Starting with 3.3.0 the checkbox will not be pre selected. You have to select the box in order to modify the bom.

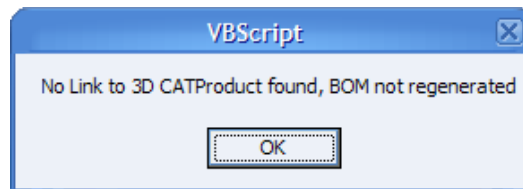


Picture 24 Modify Bom Checkbutton

If you want to hide/show the Bom table in the drawing, you can check the 2nd checkbox:



This will only work for drawings derived from a CATProduct. If the current sheet is not linked to a 3D product, the following message will pop up:



Picture 25 Bom Info message

and the bom will stay as it is.

2.4.3.10 Handling of CATIA tables

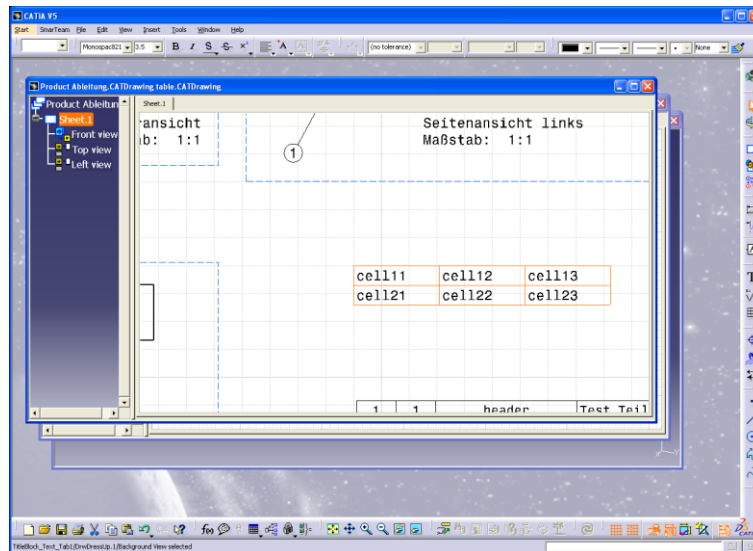
The titleblock editor allows to edit tables which have names according to the editor convention. If you place a table in your drawing and assign a component id of

TitleBlock_Text_TableName

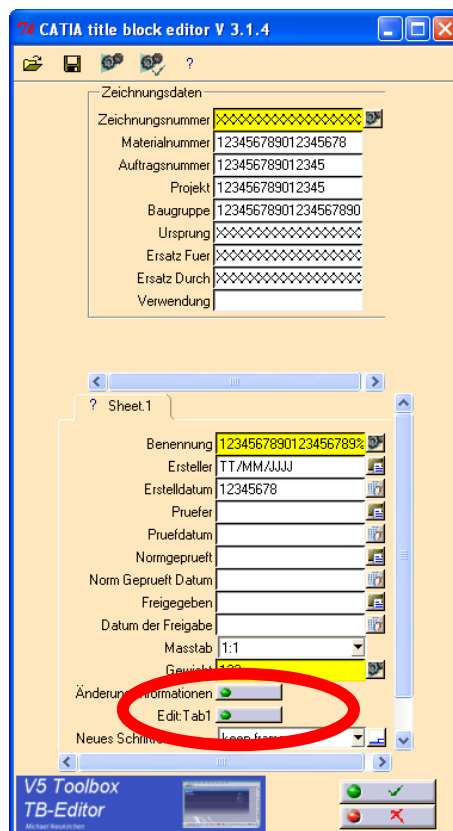
Then the editor will offer you the possibility to edit that table.

2.4.4 Sample Table

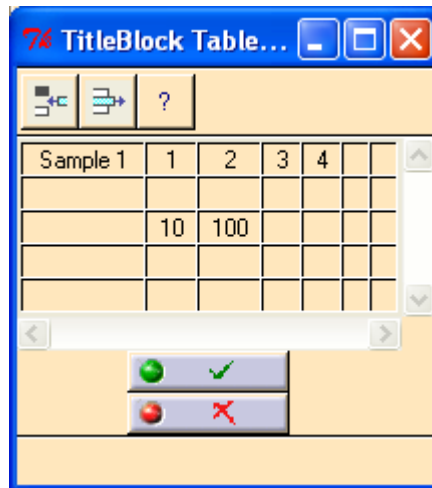
The sample below shows a 3by2 Table. The editor offers an additional button to edit the table.



Picture 26 Table in CATIA Drawing



Picture 27 Button to edit the table



Picture 28 Table Editor

The Ok Button will accept the changes, the Cancel button discard changes done in the Table Editor.

The Buttons in the upper part of the panel allow you to add or to delete a row. Alternatively you can use the right mouse button to get a pop up menu. In each case, a cell has to be selected first.



Merging of cells is not supported!

2.5 Missing fields

In case of missing title block fields in the CATIA model, the macro will display a summary of the fields missing. You should add the required fields to your model.



Picture 29: Missing Elements in the drawing frame

This sample message reports one error:

The field *TitleBlock_Text_Number* is missing in sheet *Blatt .1*

(Here you can see the German translation)

2.5.1 Change

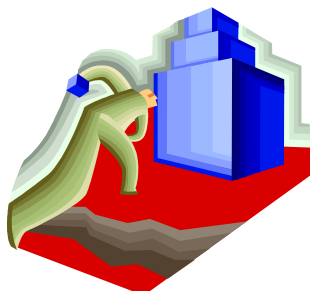
In case you erroneously had active a wrong TitleBlock the selection of Change (upper button) will pop up the dialog to select a new frame. This new frame will be loaded into the background view.

2.5.2 Cancel

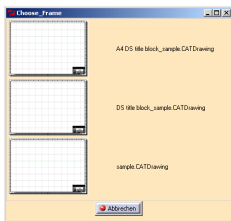

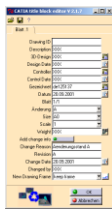
Selection of Cancel will cancel the macro. No changes will be made.

2.6 Use Cases

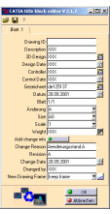
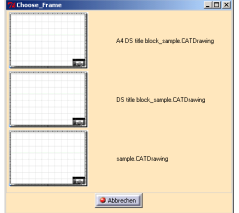

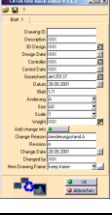
This chapter describes some use cases for the macro



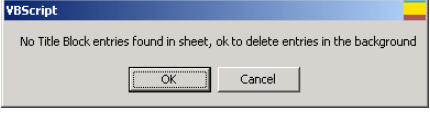
2.6.1 Empty Model

Action	Response
Call the macro	<p>Display of frame selection panel</p> 
Select Cancel	The process is aborted, the model remains empty
Select Frame	<p>The frame is copied into the background of the current sheet</p>  <p>The editor pops up</p> 
Edit data within the editor panel + ok	Data is sent to the model and is edited in the frame
Select cancel	The data from the template frame will be in the model

2.6.2 Model containing a frame

Action	Response
Call the macro	<p>The editor pops up</p> 
Select Cancel	The process is aborted, the model remains as before
Select a New frame + ok	<p>The frame selector shows up</p>  <p>The new frame is copied into the background of the current sheet</p>  <p>The editor pops up again</p> 
Edit data within the editor panel + ok	Data is sent to the model and is edited in the frame
Select cancel	The data from the template frame will be in the model

2.6.3 Model with something else in the background view

Action	Response
Call the macro	<p>An message is displayed, showing that the macro couldn't identify the geometry in the background.</p> 
Select ok	The geometry in the background is erased and the frame selection panel pops up and the sequence continues like the <i>empty model case</i>
Select cancel	The macro quits





3 Part II: Admin Guide

CATIA V5 Title Block Editor V 4.4.8 Admin Guide





3.1 Installation

3.1.1 General Client / Server installation

If you want to test the program on a Server machine, follow the instructions below.

If you want to give access to a community of users, install the macro on a server like shown below. The directory has to be accessible by the clients. The `sInstallDir` has to be the path as 'seen' from the clients.

3.1.2 National Language Support

Within the `skin.tcl` data set in the config directory, you will be able to choose the language. As of today

de (german)

en (english)

are supported. English will be the default.

The specific names of the TB fields (e.g. Materialnummer) are defined within the config data set `config\custom\standard\title_block_config.tcl`. The default panel is in German language. **If you want to use English panels, please replace the `title_block_config.tcl` by `title_block_config_en.tcl`**

3.1.3 Windows / Unix Change of the TitleBlock.Catscript file

Extract the `title_block.zip` file to an arbitrary directory. You have to change one line inside the `TitleBlock.CATScript` file to reflect the location of the files:

```
sInstallDir = "C:\data\data_actual\src\catia\title_block\lv4.4.0"
```

You can use UNC notation (e.g. [\\server\dir_sample\...](#)) or arbitrary directory names. The restriction to not using embedded blanks has been removed.

You must not use blanks in the install path!

For Windows: If you want/have to use blanks within the path name, you should use the 'good old' windows/dos convention like

```
'C:\Dassault Systemes\scripts' -> 'C:\DASSAU-1\scripts'
```

The old naming convention can be checked by using the `dir /X` command.

3.1.4 Unix

The application now (beginning with 3.1.0) uses pre-built starkits which include the necessary widgets.

Download the Unix – Support for the TitleBlock Macro from www.catiav5forum.de and place the binaries in the corresponding directory/directories. The version TB-UnixSupport_V1.1 will be necessary.

bin/intel	windows
bin/hpux	HP-HPUX
bin/irix	SGI-IRIX
bin/solaris	Sun-Solaris
bin/aix	IBM-AIX

The program has been tested under AIX. Any experiences are welcome and will be integrated.

You have to change the executable flag for the binaries, so issue a command like

chmod a+x Titleblock.exe

3.1.5 Ready to Go!

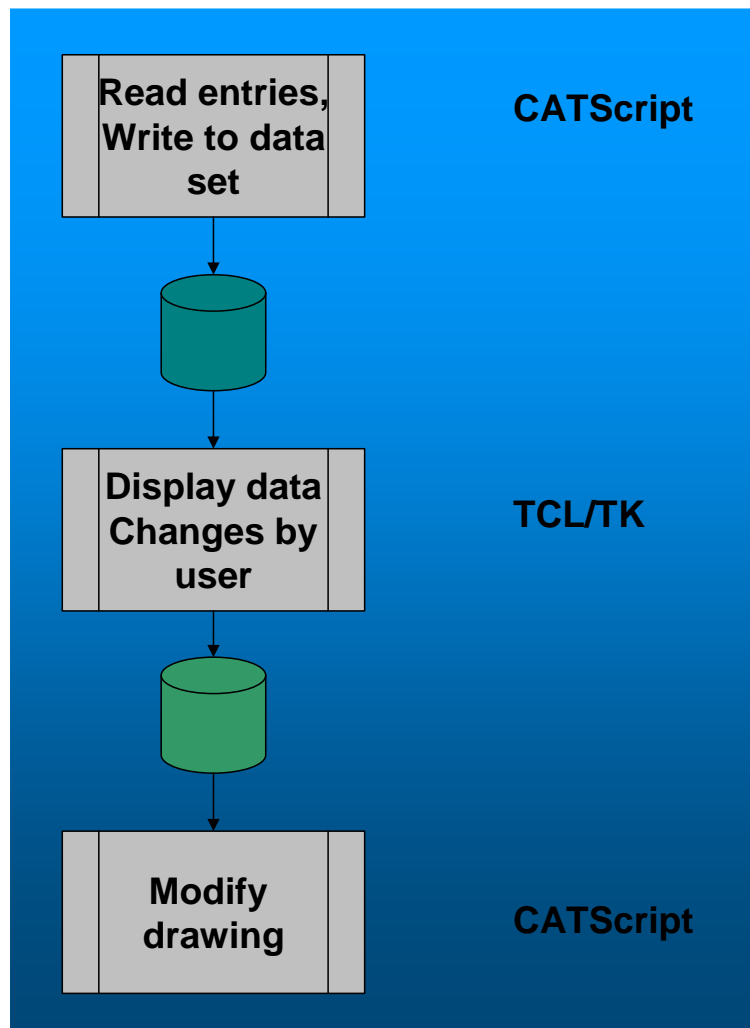
You should now be ready to go. Add the path of the macro to your Macro libraries and execute it.

3.2 How does it work?

3.2.1 In general

The macro is written in Basic Script and TCL/TK. The Basic script does the interface to CATIA, the TK part provides the GUI to enter data.

First the macro checks, if the drawing already has title block info in the background view. If so, the GUI pops up and the user can change data. If not, a selection panel for title block frames pops up. In this case, all info in the back ground is deleted and the macro copies the selected frame to the background view of the current drawing.



In case you exchange an existing frame in the drawing (e.g. replacing an A1 by an A0 size frame) the macro will erase the background completely. So ensure that you only have the frame info in the background (instatiated details – components will be preserved). If you need additional geometry individually for drawings, you can add this by details. Possibly you will have to rearrange the details after replacing the frame.

The script uses starting with 2.4.1 a brute force communication between the GUI and the CATIA script. A temp data set

cat_title_block_3.txt

is used to check if the GUI is still active. CATIA polls the whole time. If CATIA hangs after the macro, kill this data set in the temp directory.

3.2.2 CATIA documents supported

In CATIA V5 you can define drawings within CATDrawings (no wonder), CATProducts (Diagrams) and CATParts (with the help of LO1).

With release 3.2.0 the TB Macro supports

- CATDrawings
- CATProducts for Diagrams

3.2.3 SMARTEAM compatibility

If the configuration data set for SMARTEAM/TD1 is available, the macro assumes a TD1 compatibility. The CATIA SMARTEAM Integration relies on the mapping of parameters from CATIA to SMARTEAM



objects. The macro will connect the title-block entries with CATIA parameters. The program works as follows:

- Delete the parameters, which are specified in the config data set
- Create the parameters and connect them to the corresponding title block entries

This procedure allows an exchange of title block frames, because all title block entries are automatically reconnected to the corresponding parameters. Even drawings, which don't have parameters, will be provided with the parameters and can be easily interfaced with TD1. The basics are: Have title-blocks containing texts with the 'right' object names.

3.2.4 Support of Library Components

You can use CATIA libraries to place components on the background. This can be modular title blocks or geometry like welding info.

If you want to include texts in the components, which should show up in the editor, you can use modifiable texts using the same naming conventions (meaning TitleBlock_Text_...).

If you exchange the frame, the components will be copied to the new frame. A shift in x-direction will be performed according to the different horizontal sizes.

Do not use other geometry on the background view, since this geometry will be deleted, in case you will perform an exchange of the title frame geometry (e.g. A1 → A0).

3.2.5 Multiple configurations

In some cases it is necessary, that you use more than one configuration. That may happen for instance when you are part of a supply chain. These configurations can be defined and stored in the config/custom directory. Simply add an new directory which will hold your configuration files.

If you start the macro, and you already have TB-Data within your drawing, the macro checks, which config fits best to your active drawing. It compares the field definitions of the configuration data set with the entries in your drawing. In case it finds a config which has all fields defined, the macro will switch to this config.

You are able to switch configs and frames. In that case, the macro will map the existing entries in your drawing to the new field definition. **If you have defined a field on your drawing, and that field is not defined in the new configuration, the entry information will be lost.**

Starting with 4.0.0 the TB macro supports multiple configurations. However, as long as you are using one configuration – the “standard” config – the behaviour of the TB-Macro will be as in Release 3.

The default directory will be the “standard” directory . If you want to use another name, you can do so by specifying the default directory within the basics.cfg data set. This default configuration will be used for new drawing frames.

4.0.4

3.3 Customization



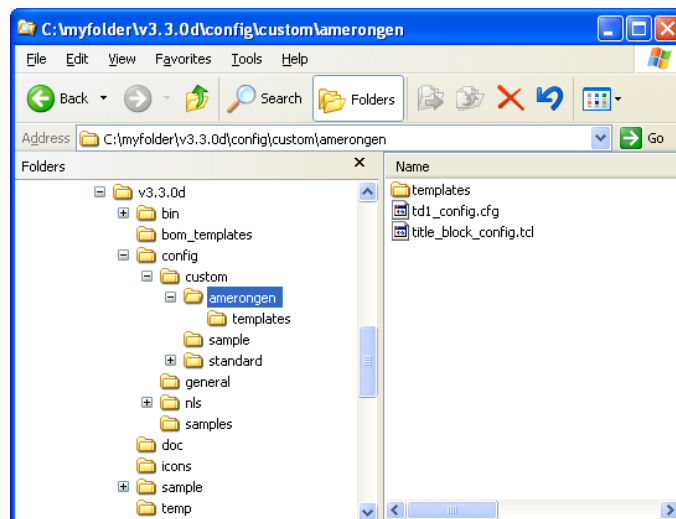
Roadmaps to configure the title block scenario. Since lots of requests have been posted, the configuration has lots of entries.

Just start with the template provided, and check if the customization described in this chapter offers some parameters to exactly tailor the titleblock macro to your needs.

3.3.1 Unload directory

The files can be unpacked anywhere. Under the unload directory you will find

Name	Type	Comment
Titleblock.CATScript	File	CATIA Macro
Bin	directory	Launcher for help, main script file, tcl scripts, executables (Windows)
Doc	directory	Documentation
Icons	directory	Pictures, used by the TCL script
Config	directory	Configuration files
Sample	directory	Sample CATIA files
config/custom/templates	directory	Templates for the drawing frames, that is a pair <ul style="list-style-type: none"> Frame_Sample.CATDrawing Frame_Sample.gif The gif will be shown, to give the user a help to select the right frame.



Picture 30 Unload structure

Within the config directory you will find general configs (mainly NLS or skin related) and the individual configs. If you only want to use one configuration, simply use the config/custom directory for your changes.

3.3.2 English Panels

If you want to use the English version, please rename the config data set

config\custom\standard\ title_block_config_en.tcl to

config\custom\standard\ title_block_config.tcl

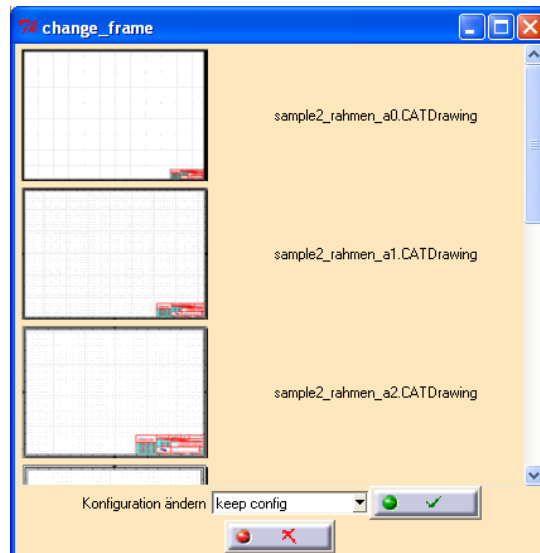
This will give you a first impression of the standard behaviour of the TB Macro with English panels. For other languages, please edit the title_block_config.tcl and use your specific expressions.

3.3.3 Add a new drawing frame

Fur sure, your company will have other drawing frames than the templates provides. In order to use your own drawing frames you have to perform the following steps:

- 1) Get the geometry from one of your existing frames (import it from dxf, ccd,).
- 2) Modify the component names of the relevant texts in your frame (see following chapter). Relevant texts are those texts, which should be edited using the titleblock macro. E.g.: The description field of your frame may have a component name like Text.154. This will have to be changed to be changed to TitleBlock_Text_Description. (Details see below).
- 3) Put the geometry into the Background (select all, cut, paste)
- 4) Put the model into the template directory
- 5) Make a picture for the drawing (capture it with Alt/Print (Alt/Druck) and use a paint program to cut and size it). Store it with the same name as the drawing, but in gif file format. If you have a drawing named "frame1.CATDrawing", you should store the picture under "frame1.gif" in the same directory.
- 6) Take one note and change the component name to Frame_Name_name_of_the_drawing. In our sample: Frame_Name_frame1. This will enable a batch exchange of the drawing frames. In my samples I use the format specifier (e.g. A4).

Now you have a new frame available and you can select it.



Picture 31: Selection of drawing frames

3.3.4 Overview

The program uses the naming convention

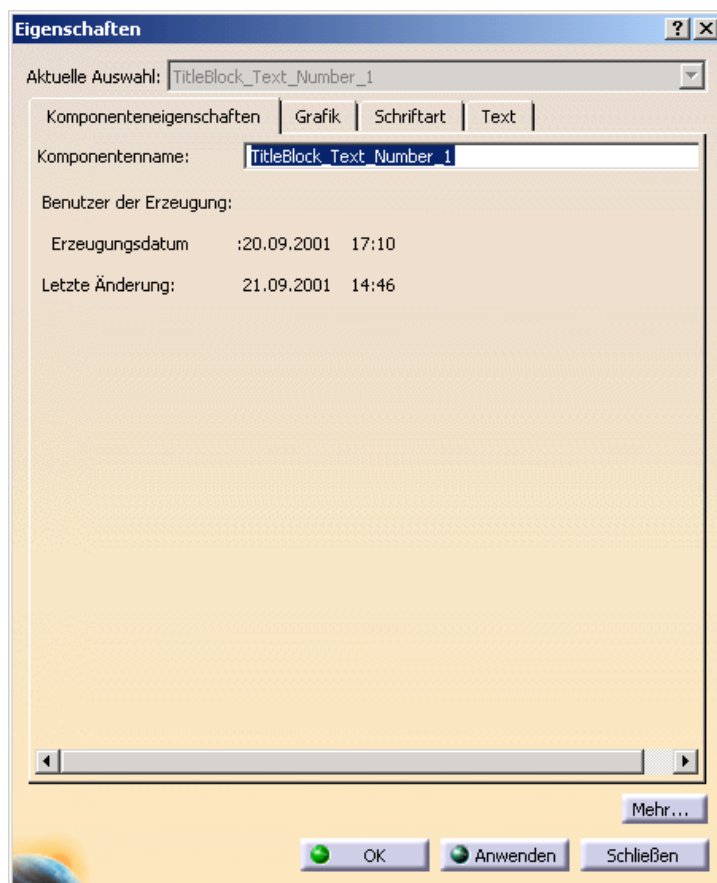
TitleBlock_Text_field

for standard fields and

RevisionBlock_Text_field_A.m

for change info fields (higher revisions will go to _B, _C and so forth, the sequence is customizable in the config data set). This must be the component names of the CATIA title block texts (change the default name via the properties contextual menu in CATIA).

The naming convention follows the Dassault sample Title Block macro (sample 2) found in the installation directory Vbscript.



Picture 32: CATIA component names

The program checks with the config/title_block_config.tcl the values found within the CATIA drawing. Within this configuration file you can specify the GUI of the macro (sequence of fields, Labels, entry-types).

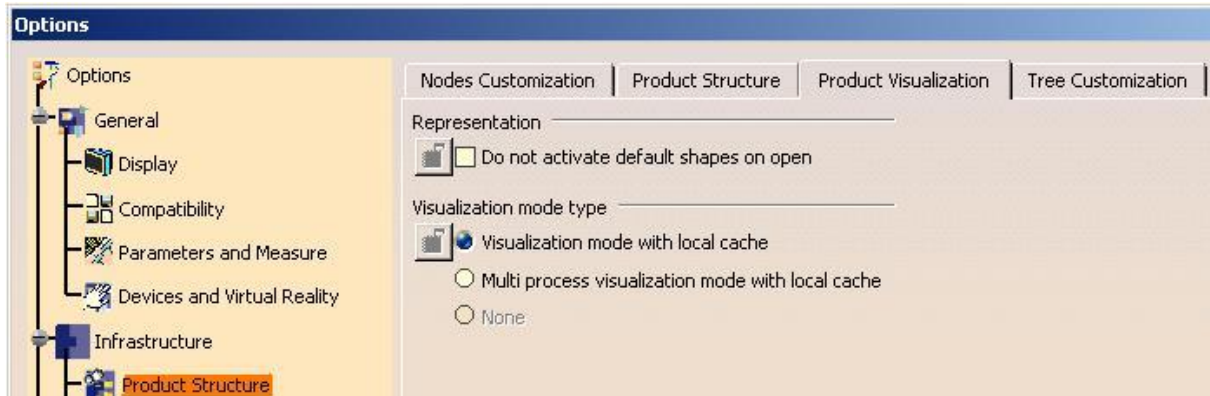
Some values from 3d are hardcoded such as:

Value in 3D Part	Name of title block entry
Number	TitleBlock_Text_Number
Title	TitleBlock_Text_Title
Revision	TitleBlock_Text_Revision
Nomenclature	TitleBlock_Text_Nomenclature
Source	TitleBlock_Text_Source
Weight	TitleBlock_Text_Weight
Volume	TitleBlock_Text_Volume
WetArea	TitleBlock_Text_WetArea
Description	TitleBlock_Text_Description
Material Path	TitleBlock_Text_Material
Name of linked 3D model incl Path	TitleBlock_Text_3DModelFullName

Name of linked 3D model	TitleBlock_Text_3DModelName
-------------------------	-----------------------------

Be aware:

the weight will not appear automatically on the title block if the Tools/Options/Product Visualization/Do not activate default shapes on open is selected.



Picture 33: Option Setting to get the Weight Info



The Macro identifies the number of revision by the field: **RevisionBlock_Text_Rev_n**, so this component names should be in the model.

Starting from 3.0.3 the prefix for the titleblock fields can be configured. This is done in the basics.cfg data set. You are no longer fixed to the TitleBlock_Text component name and can use other names, which may be given by your customer. If you specify a prefix (e.g. ABC) , the macro will check for texts like ABC_field instead of TitleBlock_Text_field (see above).

3.3.5 Configuration data sets

This chapter describes the different configuration sets for the macro.

3.3.5.1 Basic configuration – basics.cfg

Within this data set you can define the GUI behavior and the TD1 compatibility and turn on some tracing.

```
' compliance to TD1, i.e. the config data set will be read and parameters will be generated
' True / False
' Default: False

' → obsolete TD1_compliant=False
' disable tracing
Trace_Mode=none
' enable tracing: give the name of the file
' Trace_Mode=c:\titleblock_trace.txt
' Trace_Mode=/tmp/titleblock_trace.txt
' Trace_Mode=c:\titleblock_trace.txt
' The call of the GUI can be doen in foreground or in background
' foreground: CATIA will not repaint its window
' background: High CPU utilisation, CATIA will hang, if the GUI abends somehow (you have to
delete
' the temporary data set, the CATIA will continue)
' Default: background
CALL GUI_Mode=foreground
' CATIA Version String will be edited by the actual Values of Version, Release and SP Pack
'
' Variables to use: &Version
'                  &Release
'                  &SP
' CADSystem=CATIA V&VersionR&ReleaseSP&SP
CADSystem=CATIA V&VersionR&Release
' Retrieve 3D Analysis Data from 3d (set it to false and no prompting for the retrieval
' will be done
3DAnalysis=true
'
' Prefix for the component name of Texts: Default TitleBlock_Text
TBPrefix=TitleBlock_Text

' Default Configuration, will be taken if a config cannot be determined from an existing frame
DefaultConfig=standard
'
' Stream all properties with their original name (without prefix UP)
StreamAllProperties=False

' leading zeroes for sheet numbers below ten
leadingZeroes = True
```

The TD1_compliant variable is no longer necessary. If you place a TD1 config file within the config directory, the macro will assume a TD1 compatibility mode.

If you need more than one prefix – maybe you use some programs to extract data form the TB – you can also specify a list of prefices :

Sample

TBPrefix=SAP,SC5

3.3.5.2 title_block_config.tcl configuration data set

This data set can be found under the directory config.

For the title block fields we have mainly 2 types

- fields which are the same for all sheets (global_fields)
- fields, which are specific to one drawing sheet

Lines with a '#' in the first column are regarded as comments. You will find 2 examples below, showing different configurations. One small and easy to do, on more complex, showing all the customizations possible.

Start with the first one, tailor it to you needs. If you want specific customizations check the complex customization. Here you will find various options, the TB users have requested.

3.3.5.2.1 Sample 1: small config

This sample assumes, that you have stored your config in config/custom/standard an your templates in config/custom/standard/templates.

```
# catia name: id corresponds to cat_title_block_id.n
#
#
#
#
set global_fields(1)      {Number      "Zeichnungsnummer" entry}
set global_fields(2)      {Materialnummer Materialnummer entry}
set global_fields(3)      {Auftragsnummer  Auftragsnummer entry}
set global_fields(4)      {Projekt        Projekt        entry}
set global_fields(5)      {Baugruppe       Baugruppe       entry}
set global_fields(6)      {Ursprung        Ursprung        entry}
set global_fields(7)      {ErsatzFuer      "Ersatz Fuer"   entry}
set global_fields(8)      {ErsatzDurch     "Ersatz Durch"  entry}
set global_fields(9)      {Verwendung      Verwendung      entry}
set global_fields(10)     {Weight          Gewicht        entry}
set global_fields(11)     {Werkstoff       Werkstoff       entry}

set fields(1)             {Description    Benennung       entry}
set fields(2)             {Bearbeitet     Ersteller        entry}
set fields(3)             {BearbeitetDate Erstelldatum     date}
set fields(4)             {Geprueft       Pruefer          entry}
set fields(5)             {GeprueftDate   Pruefdatum       date}
set fields(6)             {Norm            Normgeprueft      entry}
set fields(7)             {NormDate       "Norm Geprueft Datum" date}
set fields(8)             {Freigegeben     Freigegeben       entry}
set fields(9)             {FreigegebenDate "Datum der Freigabe" date}
set fields(10)            {Masstab         Masstab          list}

# Date Format, dateUpper Yes will force everything to uppercase: Wed -> WED
set date_format "%d/%m/%Y"

#####
# Predefined Lists section
#####

# Define Lists for selections
# the name of the list must correlate to the name of the field
# i.e. l_CATIAName

set l_Masstab             {"1:1" "1:2.5" "1:5"}

#####
# Revision section
#####
```



```
set fields(6)           {Norm           Normgeprueft      entry}
set fields(7)           {NormDate      "Norm Geprueft Datum"  date}
set fields(8)           {Freigegeben    Freigegeben        entry}
set fields(9)           {FreigegebenDate "Datum der Freigabe"  date}
set fields(10)          {Massstab       Masstab            list}
set fields(11)          {Origin          Origin            entry}
set fields(12)          {Projection_1st_Angle Projection_1st_Angle entry}
set fields(13)          {Projection_1st_Angle Projection_3rd_Angle entry}
```

```
#####
# Extended field definitions
#####

set extFieldDef(Projekt) {"entry_width=9" "entry_alignment=right" \
                          "Mode=mandatory" "Mask=0000-0000" \
                          "Help=4711-4711")

# a text entry field with 2 lines and 20 characters width
set extFieldDef(Description) {"text_width=20" "text_height=2"}

# Date Format, dateUpper Yes will force everything to uppercase: Wed -> WED

set date_format "%d/%m/%Y"
set dateUpper "Yes"
```

```
#####
# modeMandatory can be one of:
#   none      - no field is mandatory
#   enabled   - fields are mandatory and a warning is issued if mandatory
#               fields are not filled
#   forced    - the TB macro will only offer an ok, if all mandatory
#               fields are filled
# default: none
#
# to mark a field as mandatory, you can add as a last item for the field
# the string mandatory
#
# sample:
#
# set fields(1)    {Description Benennung entry mandatory}
#
# the character, which will be seen as an unfilled field can be defined
# through the emptyChar Variable (default "-")
#####
```

```
set modeMandatory none
set emptyChar "-"
#
#####
# display extended menus, if you don't want to offer the menus, set
# the value to false - default: true
#####
set showMenuConfig true
set showMenuAllFrom3D false

#####
# Units section
#####
# define the list of fields for which a Unit conversion will be done
set u_Units {Weight}
```



```
# define the Units for each field, the first unit will be the default
set u_Weight { {kg 1} {g 1000} {mg 1000000} {t 0.001} {lb 2.204623} {oz
35.27396} {slug 0.06852177} }
# define the precision (decimal digits) for each field
set dp_Weight 2
# use the old behavior - don't display the Unit in the tb, switch to yes
# to add the unit (kg, lb, ...)
set showUnit_Weight no

#####
#   Predefined Lists section
#####

# Define Lists for selections
# the name of the list must correlate to the name of the field
# i.e. l_CATIName

set l_Massstab      {"1:1" "1:2.5" "1:5"}

#####
#   Revision section
#####

# Define the possible revisions , has to be the same convention like for
the
# Component Names of the revision entries in the CATIA titleblock

set revisions { 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 }

set change_fields(1) {Rev                "Aenderungs ID"        entry}
set change_fields(2) {Aenderung           "Beschreibung"        entry}
set change_fields(3) {Datum               "Datum der Aenderung" date}
set change_fields(4) {Name                "Name"                entry}

# Add default values to select for the Rev entry in the revision panel
# this will offer combo boxes with values from 1..15
set l_Rev          {1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 }
set revisionEnableDelete "yes"

#####
#   Default section
#####
# Add default values
# --> set cat_title_block_field_default variable

if {$tcl_platform(machine) == "intel"} then {
    set TitleBlock_Text_Bearbeitet_default $env(USERNAME)
    set TitleBlock_Text_Freigegeben_default $env(USERNAME)
    set TitleBlock_Text_Norm_default       $env(USERNAME)
    set TitleBlock_Text_Geprueft_default   $env(USERNAME)
    set RevisionBlock_Text_Name_default    $env(USERNAME)
} else {
    set TitleBlock_Text_Bearbeitet_default $env(USER)
    set TitleBlock_Text_Freigegeben_default $env(USER)
    set TitleBlock_Text_Norm_default       $env(USER)
    set TitleBlock_Text_Geprueft_default   $env(USER)
    set RevisionBlock_Text_Name_default    $env(USER)
}
}
```




```
#####
#  template section
#####
# define the directory for the CATIA Drawing Templates

set CATIA_TEMPLATES "$where_am_i/config/custom/standard/templates"

#####
#  GUI section
#####
# width for entries
set entry_width 20

#Screen size for the notebook part of the window
set screen_height 300
set screen_width 270
#Screen size for the revision fields - all revisions will be scrolled in
this area
set change_height 270
set change_width 300
#Screen size for the drawing data area
set dwgdata_height 270
set dwgdata_width 300
#Screen size for frame chooser panel (Height #entries, Width: #character, #
#      Deltay: hight per entry in pixel
set choosePanelHeight 3
set choosePanelWidth 50
set choosePanelDeltaY 110

#####
#  show, noshow the button box
#  default      : yes
#  noshow       : no
#####
set showButtonBox "yes"
#####
#  define a table for special characters (use Alt-s to activate)
#####
set specCharacterTable { "\u00D8" "\u00A9" "\u00AE" "\u00BC" "\u00BD" \
                        "\u00B5" "\u00A2" "\u00A7" "\u00B1" "\u00B2" \
                        "\u00B3"}

#####
#  For Boms: preselect Modify Bom
#  default      : yes
#####
set preselectModifyBom "yes"
#####
#  Enable call backs for TB fields
#  default      : no
#  this sample switches dependent from the valus of an Origin Field
#  the value of 2 other fields
#####
set callBackEnabled "yes"
proc tbFieldCallBack {mode sheet field number} {
    if {$field == "Origin"} then {
        if {[getTBData $sheet $field] == "EU" } then
            setTBData $sheet "Projection_1st_Angle" "X"
            setTBData $sheet "Projection_3rd_Angle" "."
        }
        if {[getTBData $sheet $field] == "USA" } then {
            setTBData $sheet "Projection_1st_Angle" "."
            setTBData $sheet "Projection_3rd_Angle" "X"
        }
    }
}
```

```

    }
  }
}
set preselectModifyBom "yes"

```

3.3.5.3 Fields

The field section describes the entries, which are specific to individual sheets. A typical example is the sheet number or the description, which will be different on every sheet.

The TB Macro support the following types of fields

- entry
- date
- list
- text

3.3.5.4 Global fields

A global field can e.g. be a CATIA **parameter** like they are used by the TD1 integration for SmarTeam. Global fields are valid for the whole drawing i.e. all sheets.

3.3.5.5 Extended field definition

Several requests have been posted to give more flexibility to the definition of fields such as

- Mandatory fields
- individual entry width
- individual alignment (left, right, center)
- mask definitions

I want to keep the 'basic' config as simple as possible. So I introduced a new variable `extFieldDef` which holds the additional information. You only need to use this variable, in case you want to specify one of the above attributes of a field.

The general syntax is

```
Set extFieldDef(fieldname) { "Attributname1=AttributValue1" "Attributname2=AttributValue2" ... }
```

3.3.5.5.1 attribute entry_width, entry_alignment

`entry_width` specifies the width of the field length as shown in the TB panel. The default will be 20. A general value for all values can be specified by the `entry_widh` variable (see below).

Sample: "entry_width=10"

The `entry_alignment` allows justification within the field. Allowed values are

- right
- left
- center

The variable `entry_alignment` will give a default for all fields. It defaults to "left".

Sample: "entry_alignment=right"

3.3.5.5.2 attribute Mask

With the help of the mask attribute, you can specify some formats, the macro will follow. For instance, if you have a part number, which follows the convention

abcd-1234-12

CATIA Title Block Editor 448.doc

Sample: “Mask=aaaa-0000-00” you can specify this mask, which will enforce 4 alpha character, a “-“, 4 numeric, a “-“ and 2 numeric characters.

Any entry of a ‘false’ character will result in a beep and the input will not be accepted. Characters like “-“, which are specified within the mask will be automatically converted (any input will be converted to this special character).

Characters for the definition of a mask:

0	Digit
A	Uppercase alphabet character
a	Lowercase alphabet character
W	Space / Blank
z	Any alphabet character
Z	Any alphabet or digit character

3.3.5.5.3 attribute Help

The help attribute will display an individual help balloon for the field. Especially, if you use the Mask attribute, you can display a sample entry for the end user.

Sample “Help=4711-1234”

3.3.5.5.4 attribute Mode

Setting the Mode to mandatory, identifies this field as a mandatory item for input. See mandatory mode below.

Sample “Mode=mandatory”

Setting the Mode to hidden, will skip the display on the panel. This is useful, if you want to import data from the 3D model and process the data using the callback mechanism and fill in other fields depending on derived values.

Assume the Partname is 0001-0088-099_Sample99 and you want to split that up into the fields Name and Number

Sample “Mode=hidden”

```
set extFieldDef(Partname) {Mode="hidden"}

proc tbFieldCallBack {mode sheet field number} {
  if {$field == "Partname"} then {
    set data [getTBData $sheet $field]
    set temp [split $data "_"]
    setTBData $sheet "Name" [lindex $temp 1]
    setTBData $sheet "Number" [lindex $temp 0]
  }
}
```

Here the Partname field will not be shown, and the values of Name and Number will be set to the values derived from Partname. You will find a description for the TCL string command in chapter 5, most likely, you will need that ☺.

3.3.5.5.5 attribute pass3dProperties

Setting the Mode to **Yes** will directly insert the properties from the 3D master part or product into the

TB-Editor. The user does not have to select the button in order to manually insert the value.

Default: No

3.3.5.5.6 attribute Default

The associated default value will be entered into the corresponding field in case the field is empty (see also emptyChar).

Default: No default value

Sample

```
set extFieldDef(Location) {Default="Frankfurt"}
```

To set the default location to the city Frankfurt.

If you want to insert the current date for date fields you have to set the default to autoInsertDate

Sample

```
set extFieldDef(GepruftDate) {Default=" autoInsertDate "}
```

3.3.5.6 Revisions

Revisions are specified in the line revisions and change_fields. If you don't want to work with revisions, comment or delete these lines.

3.3.5.6.1 Configure the revision panel

If you do not want your users to be able to delete revision entries, you can leave the configuration as is, that will be the default.

```
set revisionEnableDelete "yes"
```

Setting the value to no will hide the corresponding buttons

```
set revisionEnableDelete "no"
```

3.3.5.7 Units

Units can be specified for the Weight, WetArea and Volume fields. In the sample above you can see a configuration for the Weight field.

A sample to configure all three fields is shown below:

```
#####
# Units section
#####
# define the list of fields for which a Unit conversion will be done
set u_Units {Weight WetArea Volume}
# define the Units for each field
set u_Weight { {kg 1} {g 1000} {mg 1000000} {t 0.001} {lb 2.204623} {oz 35.27396} {slug 0.06852177} }
# define the precision (decimal digits) for each field
set dp_Weight 2
# display the Unit in the tb
set showUnit_Weight yes

set u_WetArea { {MM2 1} {CM2 0.01} {DM2 0.0001} {M2 0.000001} }
```

```
# define the precision (decimal digits) for each field
set dp_WetArea 2
# display the Unit in the tb
set showUnit_WetArea yes

set u_Volume { {MM3 1} {CM3 0.0001} {DM3 0.000001} {M3 0.000000001} }
# define the precision (decimal digits) for each field
set dp_Volume 2
# display the Unit in the tb
set showUnit_Volume yes
```

The u_Units variable holds a list of Units, which will be treated by the TB. In the sample above: Weight, WetArea and Volume. The u_ *field* holds a list of values. Each pair describes the unit and a factor to convert from the unit to other units. The dp_ *field* holds the “decimal places” and the showUnit_ *field* provides the unit in the field entry of the TB or omits the unit (set to no).

3.3.5.8 Width of entry fields

As a default, the macro uses a width of 20 characters for all entry fields. You can override this value by specifying the entry_width. As an example, we will use a width of 40 characters:

```
set entry_width 40
```

3.3.5.9 Definition of the height/width of the notebook area

In case of very complex title blocks, you can configure the height and width for the notebook area.

The definition can be done within the configuration data set.

Defaults:

screen_height	400
screen_width	270

If the notebook exceeds these values, the area will be scrolled.

3.3.5.10 Definition of the height/width of the frame chooser

You can specify the height and width for the frame chooser panel.

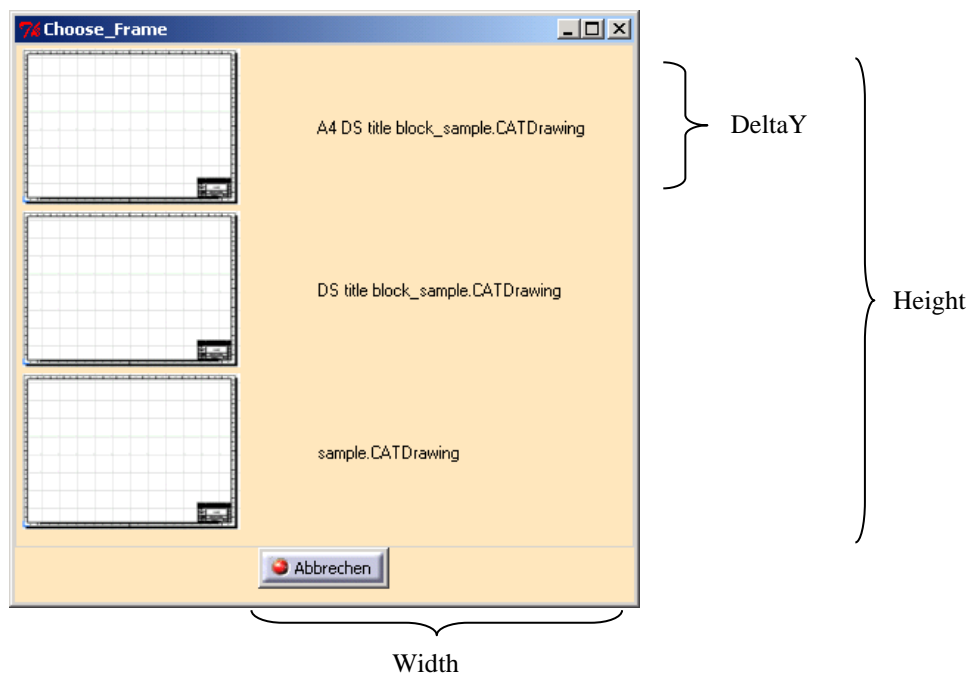
choosePanelHeight: number of frames (lines) in the chooser.

choosePanelWidth: Number of character for the drawing names

choosePanelDeltaY: Height for the frames

Defaults:

```
set choosePanelHeight 3
set choosePanelWidth 50
set choosePanelDeltaY 110
```



3.3.5.11 Date Format

For the date selector you can specify the date format. The following formats are available:

%a	Abbreviated weekday name (Mon, Tue, etc.).
%A	Full weekday name (Monday, Tuesday, etc.).
%b	Abbreviated month name (Jan, Feb, etc.).
%B	Full month name.
%c	Locale specific date and time.
%d	Day of month (01 - 31).
%H	Hour in 24-hour format (00 - 23).
%I	Hour in 12-hour format (00 - 12).
%j	Day of year (001 - 366).
%m	Month number (01 - 12).
%M	Minute (00 - 59).
%p	AM/PM indicator.
%S	Seconds (00 - 59).
%U	Week of year (00 - 52), Sunday is the first day of the week.
%w	Weekday number (Sunday = 0).
%W	Week of year (00 - 52), Monday is the first day of the week.
%x	Locale specific date format.
%X	Locale specific time format.
%y	Year without century (00 - 99).
%Y	Year with century (e.g. 1990)
%Z	Time zone name.

3.3.5.12 Areas

The last configurations define the areas of the different panels/sub panels of the GUI. If the contents of these parts are larger than the area specified, scroll bars will be inserted. Vice versa: If you encounter not wanted scrollbars when you call the macro, increase the sizes appropriately.

```
#Screen size for the revision fields - all revisions will be scrolled in
this area
set change_height 270
set change_width 300
#Screen size for the drawing data area
set dwgdata_height 270
set dwgdata_width 300
```

3.3.5.13 Special Character

You can specify a list of special characters which normally cannot be entered via the keyboard. The user can invoke the special window by using the Alt-Key together with the s key.

The characters have to be specified by their ASCII values (see appendix). If the specCharacterTable variable is not defined, the Alt-s key is disabled and no window will pop up.

```
set specCharacterTable { "\u00D8" "\u00A9" "\u00AE" "\u00BC" "\u00BD" \
"\u00B5" "\u00A2" "\u00A7" "\u00B1" "\u00B2" \
"\u00B3" }
```

3.3.5.14 Special Variables Containing CATIA infos

You can insert values, which are known by the CATIA macro, by using specific variables with predefined names.

3.3.5.14.1 CATIAVERSION

If you want to show the CATIA Version in a title block field, then use the special field Title-Block_Text_CADSystem. The string which shall be displayed can be specified in the basics.cfg data set.

The sample below used the default

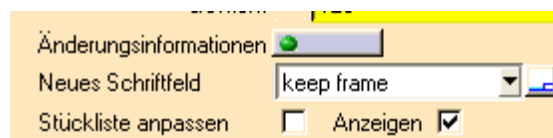
```
CADSystem=CATIA V&VersionR&Release
```

setting.

3.3.5.15 Pre-selection of the Modify Bom entry

You can switch the pre-selection of the modify bom option within then panel to on or off.

```
set preselectModifyBom "yes"
```





Picture 34 Entry panel with CATIA version and release

3.3.5.15.1 Number of Sheets

Use the TitleBlock_Text_NumSheets variable to let CATIA automatically insert the total number of drawing sheets (details not included). If the number is less than 10, a leading 0 will be used (e.g. 05).

This should be a global variable.

3.3.5.15.2 Number of Actual Sheet

Use the TitleBlock_Text _numActSheet variable to let CATIA automatically insert the actual number of the current sheet. If the number is less than 10, a leading 0 will be used (e.g. 05). The macro will give the first sheet number 1, the second number 2 and so forth – independent of the naming convention used within the tbs on top of the sheets.

This should be a sheet specific variable.

3.3.5.15.3 Scale of the sheet

Use the TitleBlock_Text _SheetScale variable to let CATIA automatically insert the main scale of the sheet. The scale will be shown as 1:Number (e.g. 1:10).

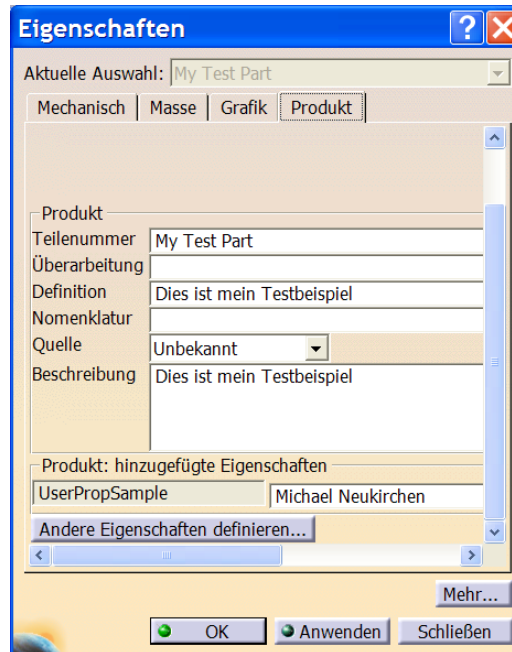
This should be a sheet specific variable.



3.3.5.15.4 Values from the 3D-Master

Besides the predefined values mentioned above, the macro will extract the user properties of the 3D-master part/product (only if the drawing is derived from a 3D master).

Sample:



Picture 35 User defined property

The macro will extract this info to a TitleBlock_Text_UPUserPropSample field. The name used is UP + Name of property.



You should use the UP prefix convention in cases, you frequently exchange models with other partners since you cannot foresee the properties which are stored in the 3D models. Since 4.0.4 you can set the variable StreamAllProperties to yes and the TB macro will pass all properties with their original name (i.e. without the UP prefix) to the user interface.

You must not use blanks in the name of a property.

If the macro detects differences regarding the actual entries in the title block field and the values from 3D, these entries will be marked with a yellow background color and the corresponding tab of the sheet is flagged with a question mark.

Picture 36 Value differs 3D and actual entry

If you press the 3D logo on the right side of the entry, the 3D value will be taken and the yellow color will change to white. If all entries are accepted, the question mark on the tab will be removed.

3.3.5.15.5 Summary of pre configured values

Here a table which shows the built in values which are directly extracted from CATIA (3D values and special values as mentioned above).

Name of title block entry	Meaning	Source
TitleBlock_Text_Number	Number	3D-Master Property
TitleBlock_Text_Title	Title	3D-Master Property
TitleBlock_Text_Revision	Revision	3D-Master Property
TitleBlock_Text_Nomenclature	Nomenclature	3D-Master Property
TitleBlock_Text_Source	Source	3D-Master Property
TitleBlock_Text_Weight	Weight	3D-Master Property
TitleBlock_Text_Volume	Volume	3D-Master Property
TitleBlock_Text_WetArea	WetArea	3D-Master Property



TitleBlock_Text_Material	Material	3D-Master Property, if no material is assigned, a blank (" ") will be shown
TitleBlock_Text_CADSystem	Version, Release, Service Pack	CATDrawing
TitleBlock_Text_NumSheets	Number of sheets in the drawing (details not included)	CATDrawing
TitleBlock_Text_numActSheet	Number of the sheet in the sequence of sheets	CATDrawing
TitleBlock_Text_SheetScale	Main scale of the sheet	CATDrawing
TitleBlock_Text_CADDWGModelName	Name of 2D Drawing	CATDrawing
TitleBlock_Text_CADDWGModel-FullName	Name of 2D Drawing	CATDrawing
TitleBlock_Text_CADDWGModelNameWithoutExtension	Name of 2D Drawing without the .CATDrawing extension	CATDrawing
TitleBlock_Text_UPName	User Property from the 3D Master	3D-Master User Property
TitleBlock_Text_3DModelFullName	Name of linked 3D model incl Path	3D-Master Property
TitleBlock_Text_3DModelName	Name of linked 3D model	3D-Master Property

3.3.5.16 Automatism for TB entries

Starting with 4.1.2 the TB macro allows a callback for TB entries. This mechanism can be activated by the "callBackEnabled" variable. Setting this variable to yes, will turn on the call back mechanism.

```
set callBackEnabled "yes"
```

Each time, a variable is modified within the GUI or filled before the GUI is shown, the tbFieldCallBack procedure will be called (see the sample below). Using this mechanism, you can reflect automatic changes of entries. The tbFieldCallBack procedure is called with 4 parameters:

mode	normal	for the main panel
sheet	revision	for the change panel of revisions
	number of sheet	0 for global parameters
		>0 for sheet specific parameters
field	name	name of the field changed
number	revision number	0 for the main panel
		>0 for the change panel

The TB fields can be accessed from the procedure by:

```
getTBData sheet field
setTBData sheet field data
getTBRData sheet field number
setTBRData sheet field number data
```

data will be the data to be set for the corresponding TB field.

Sample:

```
proc tbFieldCallBack {mode sheet field number} {
    if {$field == "Origin"} then {
        if {[getTBData $sheet $field] == "EU" } then
            setTBData $sheet "Projection_1st_Angle" "X"
            setTBData $sheet "Projection_3rd_Angle" "."
        }
        if {[getTBData $sheet $field] == "USA" } then {
            setTBData $sheet "Projection_1st_Angle" "."
            setTBData $sheet "Projection_3rd_Angle" "X"
        }
    }
}
```

The proc first checks for the field which is changed, then it sets the value of Projection_1st_Angle and Projection_3rd_Angle dependent from the value of the Origin field.

Please keep in mind: the proc is called for every change. So if the user types USA in the sample above, the proc will be called 3 times: first for the change of Origin to U, then to US then to USA.

You will find a description for the TCL string command in chapter 5, most likely, you will need that ☺.

The procedure has to be defined within the config data set.

3.3.5.17TD1 compatibility – td1_config.cfg

The file for the TD1 compatibility has to be stored in the config directory. The naming has to be:

config/custom/standard/td1_config.cfg

If this file is available, the macro will assume compatibility to SMARTEAM and will map the defined title block entries to CATIA parameters.

If set in the basics.cfg, the macro will link the given parameter to the corresponding title block field (s) if necessary.

Sample provided:

```
' Each line has the following info
' - SmarTeam Parameter Name
' - Name of text component on the sheet
' - Parameter visible in V5 tree
' - Parameter only linked on sheet 1 (e.g. as scale, material, ...)
'
'Parameter name;Text component name;visible;sheet1_only
Description;TitleBlock_Text_Description;true;false
Creation Date;TitleBlock_Text_BearbeitetDate;true;false
Approval Date;TitleBlock_Text_GeprueftDate;true;false
Originator;TitleBlock_Text_Bearbeitet;true;false
Approved By;TitleBlock_Text_Geprueft;true;false
Original Creation Date;TitleBlock_Text_BearbeitetDate;true;false
Title;TitleBlock_Text_Number;true;false
Scale;TitleBlock_Text_Massstab;true;true
```

Lines with a ' in the first column are regarded as comments.

CATIA Title Block Editor 448.doc

3.3.5.18 Changing Text Names

In some cases you will need to change some existing texts in your drawings in order to match the titleblock naming convention.

For example:

You have drawings which contain the Text Material instead of TitleBlock_Text_Material.

With the help of the mapper file, you can change the names in your drawing before the TB-editor starts the scan.

Sample:

' Text names to be changed

' old text name is changed to new text name

TitleBlock_Text_Part_Name(English) = TitleBlock_Text_Part_Name_English



3.3.5.19 Modifying the GUI – skin.tcl

In this data set you can provide your own icons or background color.

```
#
# define background color
#

set background_colour "#FFE7BD"
#
# define language, the default is English, omit
# the declaration, if you want to use English
#
set language "de"

#set background_colour "#FFFFFF"

#
# define the files for the icons here
# $where_am_i is the installation directory
#
image create photo but_def -file "$where_am_i/icons/pic_default_1.gif"
image create photo but_from3d -file "$where_am_i/icons/pic_from3d.gif"
image create photo but_calendar -file "$where_am_i/icons/pic_calendar_1.gif"
image create photo but_ok -file "$where_am_i/icons/pic_ok.gif"
image create photo but_can -file "$where_am_i/icons/pic_can.gif"
image create photo but_change -file "$where_am_i/icons/pic_change_small.gif"
image create photo but_data_copy -file "$where_am_i/icons/pic_data_copy.gif"
image create photo logo -file "$where_am_i/icons/logo.gif"
image create photo but_open -file "$where_am_i/icons/open.gif"
image create photo but_save -file "$where_am_i/icons/save.gif"
image create photo but_help -file "$where_am_i/icons/help.gif"
image create photo but_down -file "$where_am_i/icons/pic_down.gif"
image create photo but_tdl -file "$where_am_i/icons/tdl.gif"
```

If you want to use your own logo, simply exchange the logo.gif or choose another name.

3.3.5.20 NLS Support

Within the config directory, language dependant files can be supplied under an appropriate directory (e.g. de for German). The file containing the translated messages has to be named

message.cfg

The language to be selected is defined in the skin dataset (see above).

If the GUI doesn't find a message in the file, the original string (English) will be displayed. If someone supplies a file for another language – send it to me, I will add it in the distribution.

In case you want to use the English messages, set the language in the skin.tcl config file to en, sample

```
set language "en"
```

3.3.6 BOM management

For derived drawings, the script will pick up the associated 3d product from the active sheet and extract the bom to a table which has to be in the drawing. An existing bom will be updated.

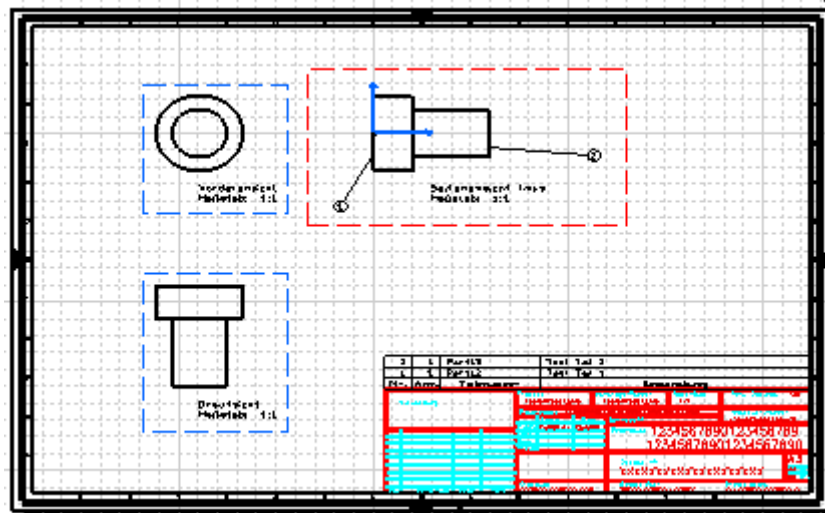
Attention: all manually changed/entered data in the table will be overridden.

3.3.6.1 Get a Bom table

Copy the Matrix for the bom (text table) from a sample drawing or library to your drawing. A sample model is provided

Product Ableitung.CATDrawing

in the sample directory.



2	1	Part13	Test Teil 2
1	1	Part12	Test Teil 1
Nr.	Anz.	Teilenummer	Beschreibung
Projekt: Auftrags-Nummer: Menge-Stück: Nr.: Blattzahl: 133			

Title block frames can be prepared in advance, to contain a bom table. E.g.: Provide a frame for A0 with a bom table and one without. One sample is provided in the template directory.

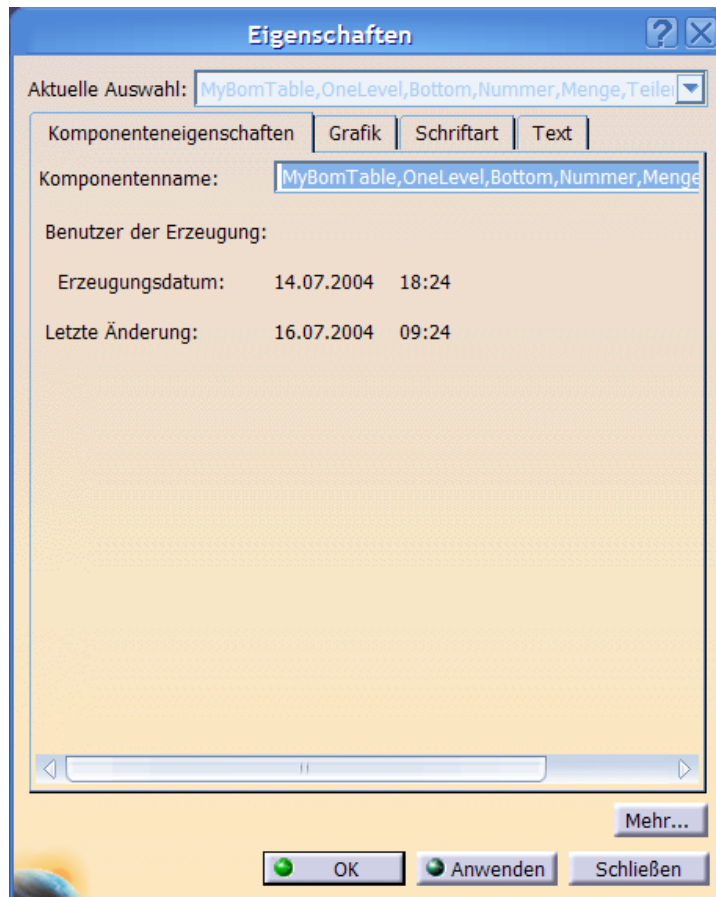
You can use all CATIA functions to build up a sample table for your bom entries. The mapping of the different columns for the bom is described in the next chapter.

The first or last row should contain the header for your bom. The placement of the bom should be up or bottom, that will keep the table in the same location even if the number of entries in the table will change. If you plan to add the bom table just above the title block, you should use the positioning bottom left (that's also done for the sample provided).

Keep playing with the options for the CATIA DrawingTable. Normally you should be able to get a layout as you want it.

3.3.6.2 Configure your Bom

The mapping of the CATIA internal bom to the table is done by defining the component name of the bom table in the right way.



Picture 37 Component Id for Bom Table

The component name is set up as follows

German Sample

MyBomTable,OneLevel,Bottom,Nummer,Menge,Teilenummer,Definition

				+---- Col: 2 nd Column for Bom
			+-----	Col: 1 st Column for Bom
		+-----	Head: Header at the Bottom or Top	
	+-----	Type: Type of Bom, Currently only OneLevel		
+-----	Tag: has to be MyBomTable			

English Sample

MyBomTable,OneLevel,Bottom,Number,Quantity,Part Number,Definition

						+----	Col: 2 nd Column for Bom	
						+-----	Col: 1 st Column for Bom	
				+-----	Head: Header at the Bottom or Top			
				+-----	Type: Type of Bom, Currently only OneLevel			
				+-----	Tag: has to be MyBomTable			

Attention: CATIA needs the PartNumber (German: Teilenummer) to be part of the BOM. If the PartNumber is omitted, you will only get blank entries in the BOM.

3.3.6.2.1 Tag – Component Id's first item

That has to be MyBomTable and is used to find the table in the drawing. The table can be placed in any view incl. background view of the drawing.

3.3.6.2.2 Type

The type is currently only OneLevel . AllLevels will be give you the recapitulation (Wiederholung) which contains all parts of the assembly.

Since the recapitulation is not sorted by the Find Numbers, the macro will do the sort. The TB macro will sort the BOM by the first column in the Bom (which normally should be the find number)

3.3.6.2.3 Head

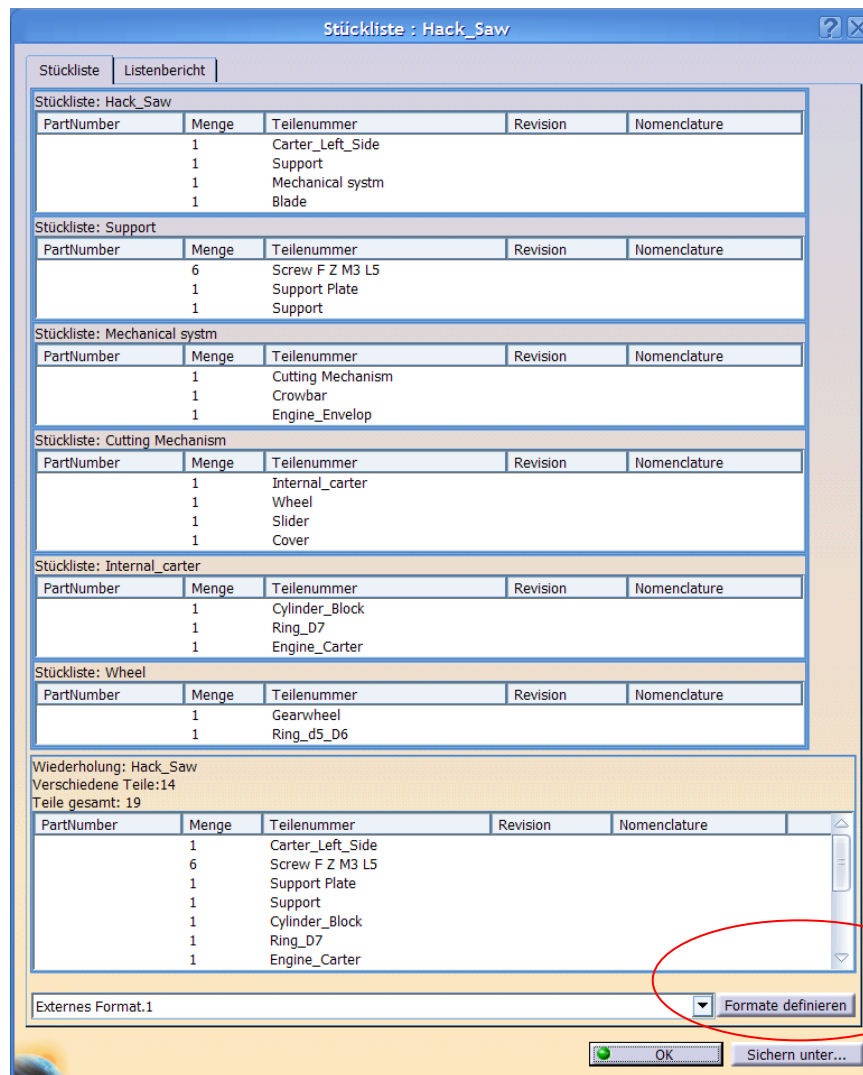
The macro will not override the first or last row of the table to preserve your headline entries for the BOM on your drawing (these headlines are normally different to the CATIA internal names for the bom entries).

Available:

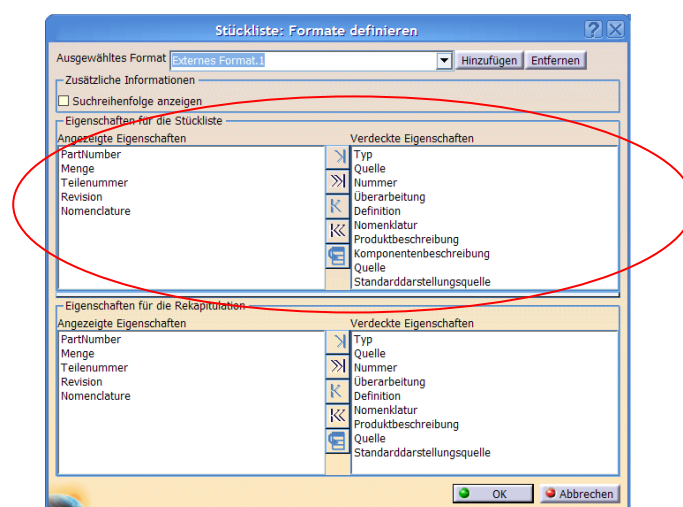
- Top
- Bottom

3.3.6.2.4 Col

The following entries have to be the exact names of the CATIA-Bom headers. These are language dependant and you can find them if you use the Analyze Menu when the CATIA product is active.



Picture 38 CATIA Bom Headers



Picture 39 CATIA Bom Properties

The items will be filled into the table just in the sequence you define in your definition of the component-id.

3.3.6.2.5 How to support Units within the BOM – Weight, WetArea, Volume

The TB macro extracts from a 3D product the weight, wetarea and volume information for the first level of its parts/products. These values can be mapped to the BOM, thus enhancing the CATIA Bom functionality.

The TB macro 'needs to know' which columns should be taken from the 3D info. This is done by placing an additional ":" and the name of the item (e.g. Weight, WetArea or Volume). In order to identify the Part, the column of the part number has to be tagged with ":Part Number".

Sample, if we want to add the weight to the BOM, we could use the sample similar as above

German

MyBomTable,OneLevel,Bottom,Nummer,Menge,Teilenummer,Definition

And modify that to

MyBomTable,OneLevel,Bottom,Nummer,Menge,Teilenummer:Part Number,Gewicht:Weight,Definition

English

MyBomTable,OneLevel,Bottom,Number,Quantity,Part Number,Definition

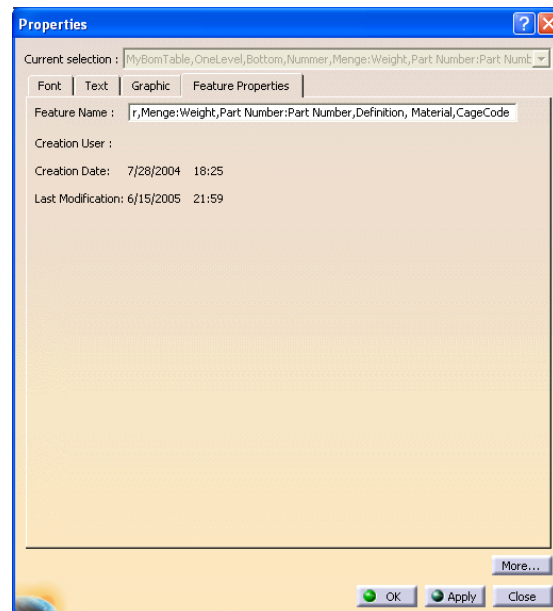
And modify that to

MyBomTable,OneLevel,Bottom,Number,Quantity,Part Number:Part Number,Weight:Weight,Definition

We added a column named Weight (Gewicht) and told the macro to insert the value of the part's weight (:Weight for both languages).

In english: the "Teilenummer" will be "Part Number", so this column has to be specified as Part Number:Part Number.

The defaults from the configuration file (Decimal Places, Default Unit, ...) will be taken into account.



Picture 40: Sample for Bom with Weight info

3.3.6.2.6 How to support user entered data

Sometimes you want to add specific data to the Bom entries. Without any special definition, this data will be erased by the macro.

However, you can specify one dedicated column, where the user can add additional data, and the macro will keep it. The naming convention for this column is User:BomData.

The TB macro checks for the PartNumber within the same line and will re-add the user defined entry. To support this feature, you have to specify the "Part Number" as above (**Part Number: Part Number** in the English sample, or **Teilenummer: Part Number** for the German Sample).

Sample:

German

MyBomTable, OneLevel, Bottom, Nummer, Menge, Teilenummer: Part Number, Gewicht: Weight, Definition, User: BomData

English

MyBomTable, OneLevel, Bottom, Number, Quantity, Part Number: Part Number, Weight: Weight, Definition, User: BomData

3.3.6.3 Override OneLevel / AllLevels

You can use the panel to allow your users to override the values defined within the BOM definition of the BOM tables on the fly.

The GUI will show the behavior defined within the component id of the BOM table of CATIA.

All Levels checked : AllLevels is set in the component id
All Levels unchecked : OneLevel is set in the component id

Checking the AllLevels box will set the AllLevels behaviour, unchecking will set the OneLevel behaviour.

If you choose AllLevels then the TB Macro will sort the BOM according to the first column in the bOM (this should usually be the first column)

3.3.6.4 What to do if something doesn't work

Check the cat_title_block_bom.txt file in the user's temp directory. Here you can see the BOM generated by CATIA.

3.3.6.5 Known Issues

CATIA doesn't provide a numbering for subassemblies. For subassemblies you won't get a find number (German: Positionsnummer) in the bom. The cell for this item will be empty.

Any idea for an automatic numbering is welcome. Currently I'm thinking to number these subassemblies with P1, P2, P3... (Product 1, product 2, ...).

3.3.7 Temporary data sets

The macro uses 4 temporary data sets

Windows

- cat_title_block.txt
- cat_title_block_2.txt
- cat_title_block_3.txt
- cat_title_block_bom.txt

in the Windows defined temporary directory

UNIX

CATIA Title Block Editor 448.doc



- /tmp/cat_title_block.txt
- /tmp/cat_title_block_2.txt
- /tmp/cat_title_block_3.txt
- /tmp/cat_title_block_bom.txt

3.3.8 Sample: Add a new entry

First: modify the CATIA sample drawing.

1. Add a new text
2. Change the component name to cat_title_block_sample1.1

Second: Modify the configuration dataset

1. Add a new item like set fields(10) {sample1 "My sample field" entry}
2. If applicable, add lists or default entries like in the sample configuration file.

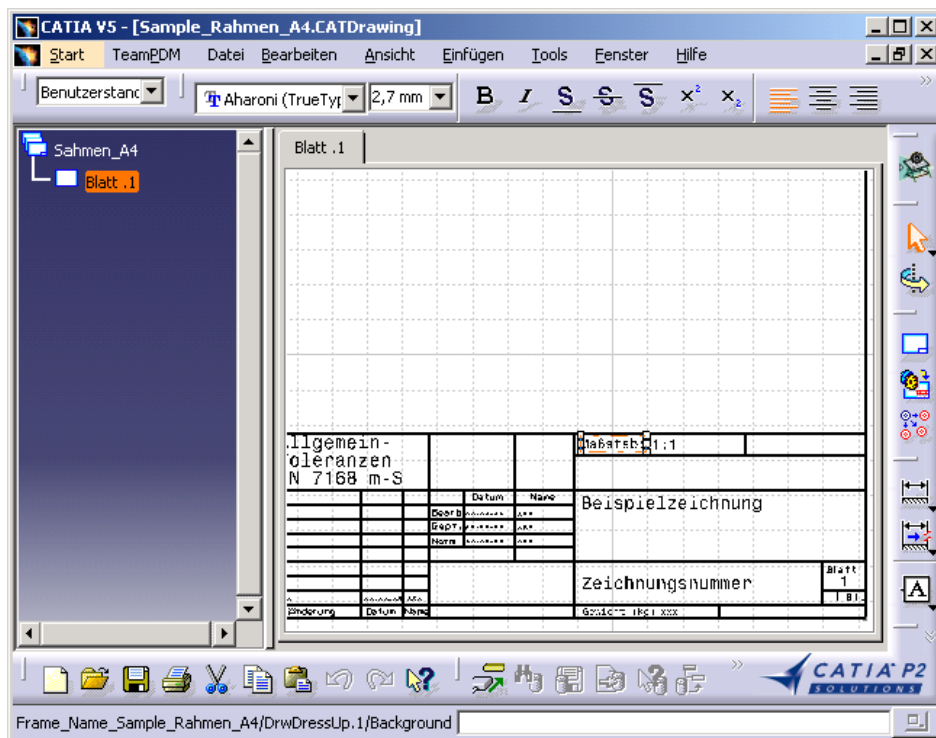
3.4 Batch exchange of title blocks

Some companies use different drawing frames e.g. in case of subcontracting. The exchange of drawing frames in batch is possible (but not yet supported).

To be able to exchange drawing sets automatically, a name of the drawing frame must be stored somehow in the model.

Proposal at the moment: Take one text and give it the component name (see previous chapter) of
Frame_Name_sample_drawing_frame_a0

Where *Frame_Name_* is the identifier for the program and the following represents the name of the CATIA Drawing (here the frame comes from: sample_drawing_frame_a0.CATDrawing).



3.5 Upgrades

It should be easy to upgrade the TB-macro. I use "should", because I cannot test all the configs you have. In case of problems, drop me a note.

The methodology to upgrade an existing customization to a higher version of the TB-macro is as follows:

Just copy the files you modified into the new config directory. These files usually are:

- basics.cfg
- skin.tcl
- title_block_config.tcl
- td1_config.cfg

and the old template directory (where you place your customized frames) to the new template directory.

If you have made changes to the NLS support, you can use you customized file again. You will find the new messages of the new version at the end of the message file. Edit them and add tem to your configuration.

3.5.1 Upgrade to version 4

I changed the directory structure of the macro. This will allow different configurations. Place your config data sets into the:

config/custom/standard

copy your template directory to config/custom/standard and edit the line with the CATIA_TEMPLATE variable to \$where_am_i/config/custom/standard/templates.

If you want to add additional configurations, create a new directory (directory nam = configruaiton name) and store your data sets in this directory.

If you want to change the name for your standard configuration directory, you can specify the name within the basics.cfg data set.

3.6 Error, Enhancement reporting

Bad news: I have to admit, that the program sometimes has one or two errors ☺

Since the site of the forum had been hacked, the wikis are gone ☹

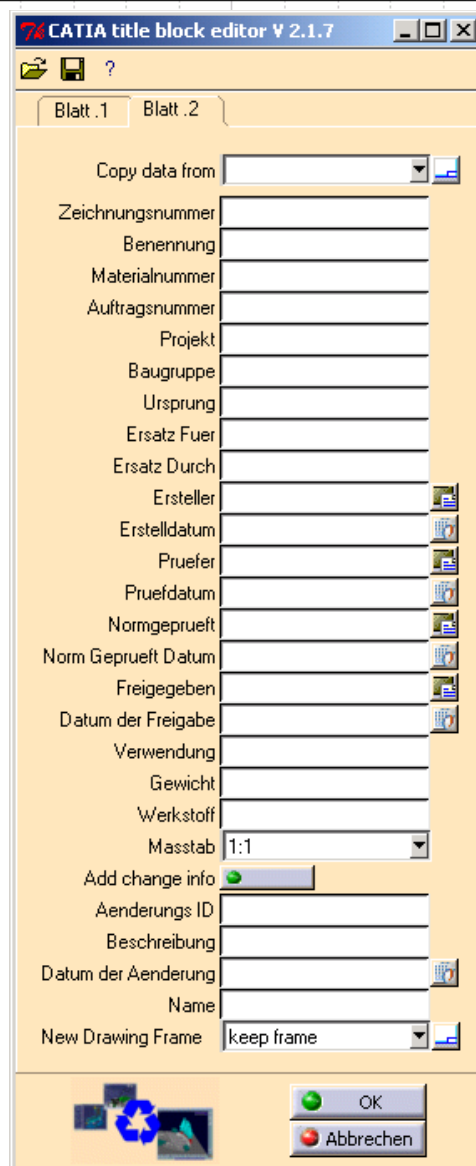
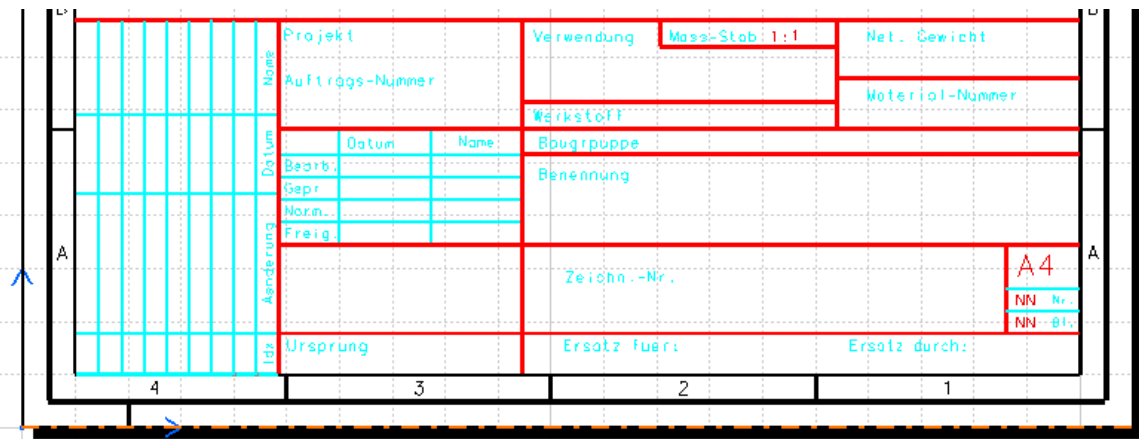
To make it easier for you, I opened 2 sites in the CATIV5 wiki, where you can post errors and enhancements.

<http://www.catiawiki.de/index.php?title=Enhancements> and

<http://www.catiawiki.de/index.php?title=Errors>

check here, if you find a bug — maybe somebody already had the same problem — and maybe, I've already analyzed or even solved it.

Send me a mail.



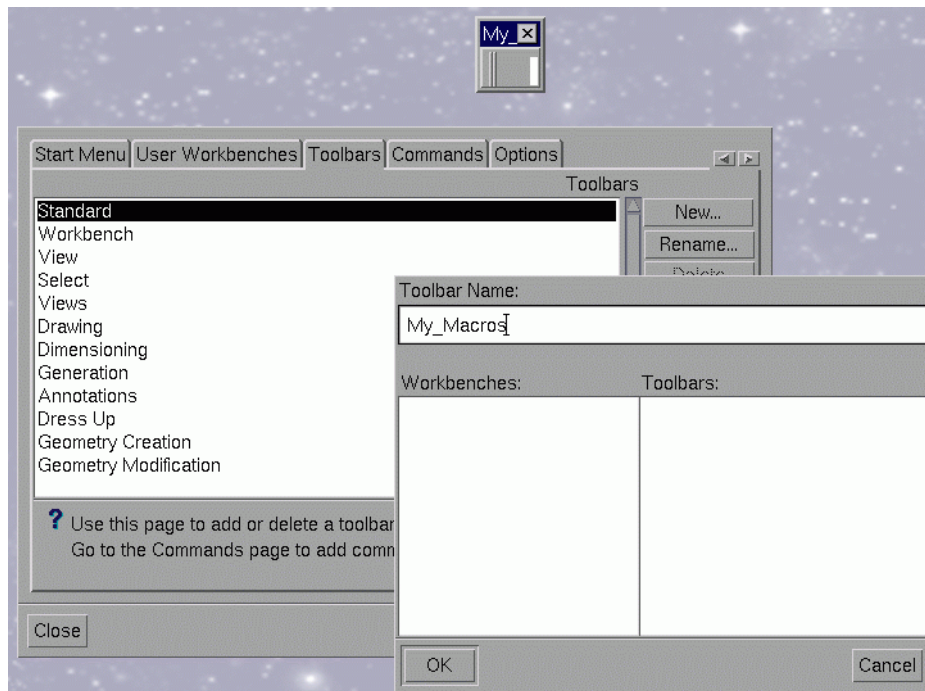


5 Part III: Appendix

5.1 How to use a button to call a CATIA macro

This paragraph shows how to create a button in your Workbench to start a CATScript macro directly.

1. Start CATIA V5 and open the workbench in which the Icon should appear.
2. Open the **Tools/Customize** menu tabs
3. Select the **Toolbars Tab**
4. Select **New** button and insert a title name for this toolbar
A small empty toolbar box is created and shows your new title

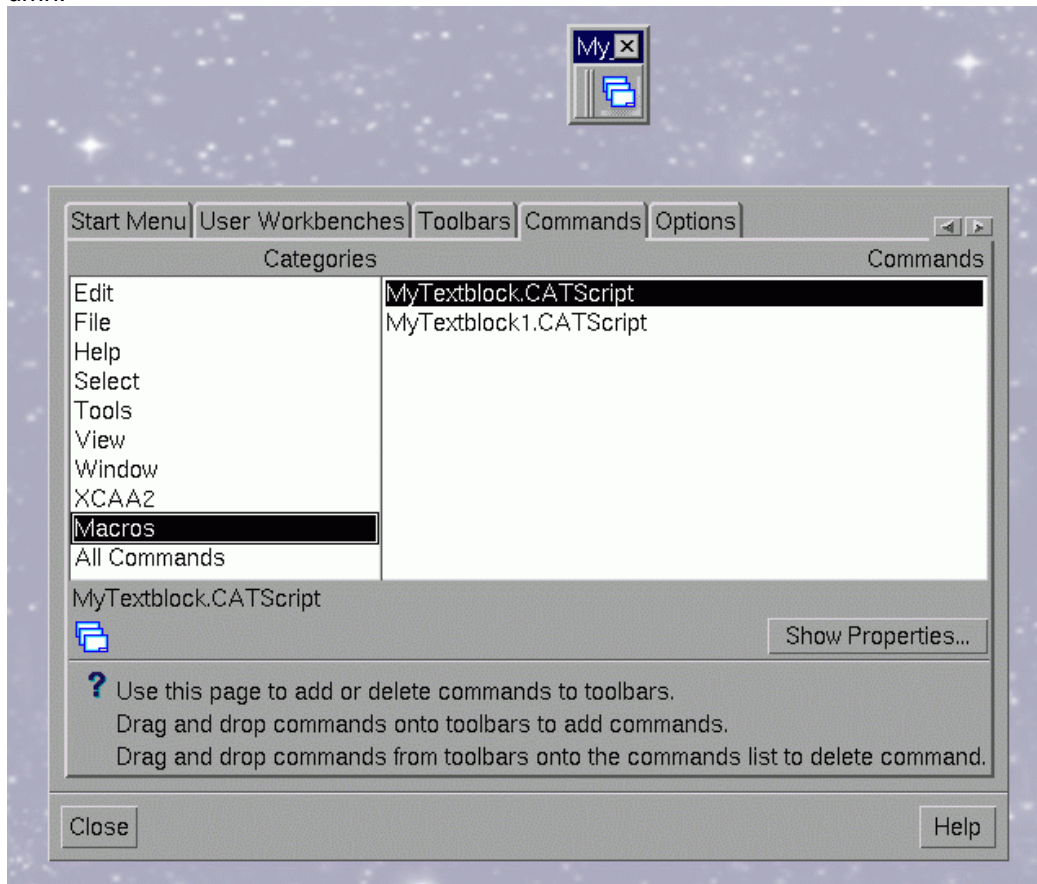


Picture 42 Toolbar definition

5. Press **OK** and switch to the **Commands** tab



6. Select **Macro** in the right column now all the known macros will be displayed in the left column.



Picture 43 Icon definition

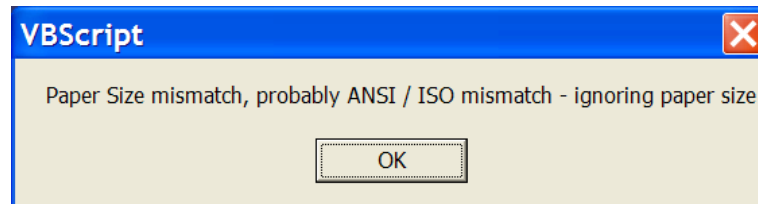
7. Select the macro which should be used in the Toolbar
If you wish to change the Icon select **Show Properties...** and use the **Icons** button for selecting an other icon from the list.
8. Drag and drop the CATScript into the empty toolbar which you created before
9. Drag and drop the toolbar to the CATIA V5 menu bar where the new icon should appear when the Workbench is opened. (In this example we used the bottom menu bar)

5.2 Known issues

5.2.1 PaperSize mismatch

If you have a drawing in ANSI format and the templates are in ISO format (and vice versa), the macro cannot automatically copy the right PaperSize to the drawing.

You will get the following message



Picture 44 Paper Size Mismatch

5.2.2 Geometry in the background

Do not place any geometry other than the title frame in the background. Since the macro pastes a background from a template drawing into the current drawing, **this geometry will be lost.**

As a workaround, you can use Details/Dittos. These Details will be placed again by the program. You possibly will have to rearrange the details.

5.3 Hints and Tips

Hints and tips from users of the title block macro.

5.3.1 How to use several Notes with one title block entry

If you have text that appears twice like my companies part number it appears once in the title block and once along the border of the drawing. You want the Title Block Program to edit both with one change. This is how you can get CATIA and the Title Block Program to change the text in both places with changing only one space in the entry panel.

CATIA will not allow two texts's to have the same name if you do it manually by changing the properties. What I have done to get around this is once the part number in the title block properties has been named correctly. I used CATIA's translate copy option to position it along the border. CATIA will allow this and now you have two texts with the same named properties. Now when you run the Title Block Program and you edit the part number in the entry panel it will change the text in both places. I have given you a part number example but you can do it with any text that appears twice and you only want to edit it once.

5.3.2 Using the CATIA DS title block macro environment

Some hints for programmer using the macro environment of CATIA to supply Title Block Macros similar to the DS examples. This macro however is designed to be used as a standard macro (Tools/Macro), it will not work in that environment.

5.3.2.1 The macro hangs somehow or does nothing

Within the title block macro environment you will not get any errors from the VB environment. Syntax errors, erroneous methods and so on will not be reported.

Develop the macro within the normal Tools/Macro environment.

5.3.2.2 The published methods are not selectable

If you select the Background logo and you don't find your methods.

CATIA scans the source first to find the published methods. These methods begin with

CATDrw_

If you don't see your methods, or a pop up with an exclamation mark appears, the code is probably not syntactically ok. To check this, call the macro with the normal Tools/Macro functionality.

5.3.2.3 I don't see any macros within the macro environment

Ensure, that the directory for the macros is defined correctly. V5R7 ships as of SP4 with \\VBScript\\FrameTitleBlock and that will not work!

5.4 Copyright Notice, Terms and Conditions



Copyright (c) 2007, 2008, 2009, 2010 M. Neukirchen. You can use the program according to the GPL 3 license.

5.4.1 GNU GENERAL PUBLIC LICENSE

Version 3, 29 June 2007

Copyright © 2007 Free Software Foundation, Inc. <<http://fsf.org/>>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

5.4.2 Preamble

The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program--to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified ver-

sions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

5.4.3 TERMS AND CONDITIONS

5.4.3.1 0. Definitions.

“This License” refers to version 3 of the GNU General Public License.

“Copyright” also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

“The Program” refers to any copyrightable work licensed under this License. Each licensee is addressed as “you”. “Licensees” and “recipients” may be individuals or organizations.

To “modify” a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a “modified version” of the earlier work or a work “based on” the earlier work.

A “covered work” means either the unmodified Program or a work based on the Program.

To “propagate” a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To “convey” a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays “Appropriate Legal Notices” to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to

view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

5.4.3.2 1. Source Code.

The “source code” for a work means the preferred form of the work for making modifications to it. “Object code” means any non-source form of a work.

A “Standard Interface” means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The “System Libraries” of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A “Major Component”, in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The “Corresponding Source” for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

5.4.3.3 2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

5.4.3.4 3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

5.4.3.5 4. Conveying Verbatim Copies.

You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

5.4.3.6 5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

- a) The work must carry prominent notices stating that you modified it, and giving a relevant date.
- b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to “keep intact all notices”.
- c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.
- d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an “aggregate” if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.



5.4.3.7 6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

- a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.
- b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.
- c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.
- d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.
- e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A “User Product” is either (1) a “consumer product”, which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, “normally used” refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

“Installation Information” for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed un-

der this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

5.4.3.8 7. Additional Terms.

“Additional permissions” are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

- a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or
- b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or
- c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or
- d) Limiting the use for publicity purposes of names of licensors or authors of the material; or
- e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or
- f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered “further restrictions” within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you

may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

5.4.3.9 8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

5.4.3.109. Acceptance Not Required for Having Copies.

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

5.4.3.1110. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An “entity transaction” is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in inter-

est had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

5.4.3.1211. Patents.

A “contributor” is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's “contributor version”.

A contributor's “essential patent claims” are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, “control” includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a “patent license” is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To “grant” such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. “Knowingly relying” means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is “discriminatory” if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you

are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

5.4.3.1312. No Surrender of Others' Freedom.

If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

5.4.3.1413. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

5.4.3.1514. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License “or any later version” applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.



5.4.3.1615. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

5.4.3.1716. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

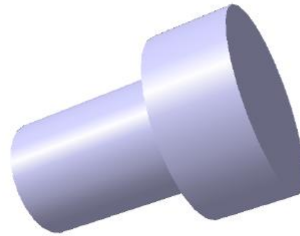
5.4.3.1817. Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS



5.5 Sample Product



5.6 ASCII Table

DEC	HEX	CHAR	DEC	HEX	CHAR	DEC	HEX	CHAR	DEC	HEX	CHAR
0	00	NUL	32	20	SP	64	40	@	96	60	`
1	01	SOH	33	21	!	65	41	A	97	61	a
2	02	STX	34	22	"	66	42	B	98	62	b
3	03	ETX	35	23	#	67	43	C	99	63	c
4	04	EOT	36	24	\$	68	44	D	100	64	d
5	05	ENQ	37	25	%	69	45	E	101	65	e
6	06	ACK	38	26	&	70	46	F	102	66	f
7	07	BEL	39	27	'	71	47	G	103	67	g
8	08	BS	40	28	(72	48	H	104	68	h
9	09	HT	41	29)	73	49	I	105	69	i
10	0A	LF	42	2A	*	74	4A	J	106	6A	j
11	0B	VT	43	2B	+	75	4B	K	107	6B	k
12	0C	FF	44	2C	,	76	4C	L	108	6C	l
13	0D	CR	45	2D	-	77	4D	M	109	6D	m
14	0E	SO	46	2E	.	78	4E	N	110	6E	n
15	0F	SI	47	2F	/	79	4F	O	111	6F	o
16	10	DLE	48	30	0	80	50	P	112	70	p
17	11	DC1	49	31	1	81	51	Q	113	71	q
18	12	DC2	50	32	2	82	52	R	114	72	r
19	13	DC3	51	33	3	83	53	S	115	73	s
20	14	DC4	52	34	4	84	54	T	116	74	t
21	15	NAK	53	35	5	85	55	U	117	75	u
22	16	SYN	54	36	6	86	56	V	118	76	v
23	17	ETB	55	37	7	87	57	W	119	77	w
24	18	CAN	56	38	8	88	58	X	120	78	x
25	19	EM	57	39	9	89	59	Y	121	79	y
26	1A	SUB	58	3A	:	90	5A	Z	122	7A	z
27	1B	ESC	59	3B	;	91	5B	[123	7B	{
28	1C	FS	60	3C	<	92	5C	\	124	7C	
29	1D	GS	61	3D	=	93	5D]	125	7D	}
30	1E	RS	62	3E	>	94	5E	^	126	7E	~
31	1F	US	63	3F	?	95	5F	_	127	7F	DEL



DEC	HEX	CHAR	DEC	HEX	CHAR	DEC	HEX	CHAR	DEC	HEX	CHAR
128	80		160	A0	NBSP	192	C0	À	224	E0	à
129	81		161	A1	ı	193	C1	Á	225	E1	á
130	82		162	A2	¢	194	C2	Â	226	E2	â
131	83		163	A3	£	195	C3	Ã	227	E3	ã
132	84		164	A4	¤	196	C4	Ä	228	E4	ä
133	85		165	A5	¥	197	C5	Å	229	E5	å
134	86		166	A6	¦	198	C6	Æ	230	E6	æ
135	87		167	A7	§	199	C7	Ç	231	E7	ç
136	88		168	A8	¨	200	C8	È	232	E8	è
137	89		169	A9	©	201	C9	É	233	E9	é
138	8A		170	AA	ª	202	CA	Ê	234	EA	ê
139	8B		171	AB	«	203	CB	Ë	235	EB	ë
140	8C		172	AC		204	CC	Ì	236	EC	ì
141	8D		173	AD	-	205	CD	Í	237	ED	í
142	8E		174	AE	®	206	CE	Î	238	EE	î
143	8F		175	AF	—	207	CF	Ï	239	EF	ï
144	90		176	B0	°	208	D0	Ð	240	F0	ð
145	91		177	B1	±	209	D1	Ñ	241	F1	ñ
146	92		178	B2	²	210	D2	Ò	242	F2	ò
147	93		179	B3	³	211	D3	Ó	243	F3	ó
148	94		180	B4	´	212	D4	Ô	244	F4	ô
149	95		181	B5	µ	213	D5	Õ	245	F5	õ
150	96		182	B6		214	D6	Ö	246	F6	ö
151	97		183	B7	·	215	D7	×	247	F7	÷
152	98		184	B8	¸	216	D8	Ø	248	F8	ø
153	99		185	B9	¹	217	D9	Ù	249	F9	ù
154	9A		186	BA	º	218	DA	Ú	250	FA	ú
155	9B		187	BB	»	219	DB	Û	251	FB	û
156	9C		188	BC	¼	220	DC	Ü	252	FC	ü
157	9D		189	BD	½	221	DD	Ý	253	FD	ý
158	9E		190	BE	¾	222	DE	Þ	254	FE	þ
159	9F		191	BF	¿	223	DF	ß	255	FF	ÿ

5.7 TCL String command

NAME

string - Manipulate strings

SYNOPSIS

string *option arg ?arg ...?*

DESCRIPTION

string bytlength string

string compare *?-nocase? ?-length int? string1 string2*

string equal *?-nocase? ?-length int? string1 string2*

string first *string1 string2 ?startIndex?*

string index *string charIndex*

integer

end

end-integer

string is class *?-strict? ?-failindex varname? string*

alnum

alpha

ascii

boolean

control

digit

double

false



[graph](#)
[integer](#)
[lower](#)
[print](#)
[punct](#)
[space](#)
[true](#)
[upper](#)
[wordchar](#)
[xdigit](#)
[string last string1 string2 ?startIndex?](#)
[string length string](#)
[string map ?-nocase? charMap string](#)
[string match ?-nocase? pattern string](#)
[*](#)
[?](#)
[\[chars\]](#)
[\x](#)
[string range string first last](#)
[string repeat string count](#)
[string replace string first last ?newstring?](#)
[string tolower string ?first? ?last?](#)
[string totitle string ?first? ?last?](#)
[string toupper string ?first? ?last?](#)
[string trim string ?chars?](#)
[string trimleft string ?chars?](#)
[string trimright string ?chars?](#)
[string wordend string charIndex](#)
[string wordstart string charIndex](#)
[SEE ALSO](#)
[KEYWORDS](#)

NAME

string - Manipulate strings

SYNOPSIS

string *option arg ?arg ...?*

DESCRIPTION

Performs one of several **string** operations, depending on *option*. The legal *options* (which may be abbreviated) are:

string bytlength *string*

Returns a decimal **string** giving the number of bytes used to represent *string* in memory. Because UTF-8 uses one to three bytes to represent Unicode characters, the byte length will not be the same as the character length in general. The cases where a script cares about the byte length are rare. In almost all cases, you should use the **string length** operation. Refer to the [Tcl_NumUffChars](#) manual entry for more details on the UTF-8 representation.

string compare *?-nocase? ?-length int? string1 string2*

Perform a character-by-character comparison of strings *string1* and *string2*. Returns -1, 0, or 1, depending on whether *string1* is lexicographically less than, equal to, or greater than *string2*. If **-length** is specified, then only the first *length* characters are used in the comparison. If **-length** is negative, it is ignored. If **-nocase** is specified, then the strings are compared in a case-insensitive manner.

string equal *?-nocase? ?-length int? string1 string2*

Perform a character-by-character comparison of strings *string1* and *string2*. Returns 1 if *string1* and *string2* are identical, or 0 when not. If **-length** is specified, then only the first *length* characters are used in the comparison. If **-length** is negative, it is ignored. If **-nocase** is specified, then the strings are compared in a case-insensitive manner.

string first *string1 string2 ?startIndex?*

Search *string2* for a sequence of characters that exactly match the characters in *string1*. If found, return the index of the first character in the first such match within *string2*. If not found, return -1. If *startIndex* is specified (in any of the forms accepted by the **index** method), then the search is constrained to start with the character in *string2* specified by the index. For example,

```
string first a 0a23456789abcdef 5
```

will return **10**, but

string first a 0123456789abcdef 11

will return -1.

string index **string** *charIndex*

Returns the *charIndex*'th character of the **string** argument. A *charIndex* of 0 corresponds to the first character of the **string**. *charIndex* may be specified as follows:

integer

The char specified at this integral index

end

The last char of the **string**.

end-integer

The last char of the **string** minus the specified integer offset (e.g. **end-1** would refer to the "c" in "abcd").

If *charIndex* is less than 0 or greater than or equal to the length of the **string** then an empty **string** is returned.

string is *class* ?-**strict**? ?-**failindex** *varname*? **string**

Returns 1 if **string** is a valid member of the specified character class, otherwise returns 0. If **-strict** is specified, then an empty **string** returns 0, otherwise and empty **string** will return 1 on any class. If **-failindex** is specified, then if the function returns 0, the index in the **string** where the class was no longer valid will be stored in the variable named *varname*. The *varname* will not be set if the function returns 1. The following character classes are recognized (the class name can be abbreviated):

alnum

Any Unicode alphabet or digit character.

alpha

Any Unicode alphabet character.

ascii

Any character with a value less than \u0080 (those that are in the 7-bit ascii range).

boolean

Any of the forms allowed to [Tcl_GetBoolean](#).

control

Any Unicode control character.

digit

Any Unicode digit character. Note that this includes characters outside of the [0-9] range.

double

Any of the valid forms for a double in Tcl, with optional surrounding whitespace. In case of under/overflow in the value, 0 is returned and the *varname* will contain -1.

false

Any of the forms allowed to [Tcl_GetBoolean](#) where the value is false.

graph

Any Unicode printing character, except space.

integer

Any of the valid forms for an integer in Tcl, with optional surrounding whitespace. In case of under/overflow in the value, 0 is returned and the *varname* will contain -1.

lower

Any Unicode lower case alphabet character.

print

Any Unicode printing character, including space.

punct

Any Unicode punctuation character.

space

Any Unicode space character.

true

Any of the forms allowed to [Tcl_GetBoolean](#) where the value is true.

upper

Any upper case alphabet character in the Unicode character set.

wordchar

Any Unicode word character. That is any alphanumeric character, and any Unicode connector punctuation characters (e.g. underscore).

xdigit

Any hexadecimal digit character ([0-9A-Fa-f]).



In the case of **boolean**, **true** and **false**, if the function will return 0, then the *varname* will always be set to 0, due to the varied nature of a valid boolean value.

string **last** *string1 string2 ?startIndex?*

Search *string2* for a sequence of characters that exactly match the characters in *string1*. If found, return the index of the first character in the last such match within *string2*. If there is no match, then return -1. If *startIndex* is specified (in any of the forms accepted by the **index** method), then only the characters in *string2* at or before the specified *startIndex* will be considered by the search. For example,

string **last** a 0a23456789abcdef 15

will return 10, but

string **last** a 0a23456789abcdef 9

will return 1.

string **length** *string*

Returns a decimal **string** giving the number of characters in **string**. Note that this is not necessarily the same as the number of bytes used to store the **string**.

string **map** *?-nocase? charMap string*

Replaces characters in **string** based on the key-value pairs in *charMap*. *charMap* is a list of *key value key value ...* as in the form returned by **array get**. Each instance of a key in the **string** will be replaced with its corresponding value. If **-nocase** is specified, then matching is done without regard to case differences. Both *key* and *value* may be multiple characters. Replacement is done in an ordered manner, so the key appearing first in the list will be checked first, and so on. **string** is only iterated over once, so earlier key replacements will have no affect for later key matches. For example,

string **map** {abc 1 ab 2 a 3 1 0} 1abcaabababababc

will return the **string** 01321221.

string **match** *?-nocase? pattern string*

See if *pattern* matches **string**; return 1 if it does, 0 if it doesn't. If **-nocase** is specified, then the pattern attempts to match against the **string** in a case insensitive manner. For the two strings to match, their contents must be identical except that the following special sequences may appear in *pattern*:

*

Matches any sequence of characters in **string**, including a null **string**.

?

Matches any single character in **string**.

[*chars*]

Matches any character in the set given by *chars*. If a sequence of the form *x-y* appears in *chars*, then any character between *x* and *y*, inclusive, will match. When used with **-nocase**, the end points of the range are converted to lower case first. Whereas {[A-z]} matches '_' when matching case-sensitively ('_' falls between the 'Z' and 'a'), with **-nocase** this is considered like {[A-Za-z]} (and probably what was meant in the first place).

x

Matches the single character *x*. This provides a way of avoiding the special interpretation of the characters **?[\]* in *pattern*.

string **range** *string first last*

Returns a range of consecutive characters from **string**, starting with the character whose index is *first* and ending with the character whose index is *last*. An index of 0 refers to the first character of the **string**. *first* and *last* may be specified as for the **index** method. If *first* is less than zero then it is treated as if it were zero, and if *last* is greater than or equal to the length of the **string** then it is treated as if it were **end**. If *first* is greater than *last* then an empty **string** is returned.

string **repeat** *string count*

Returns **string** repeated *count* number of times.

string **replace** *string first last ?newstring?*

Removes a range of consecutive characters from **string**, starting with the character whose index is *first* and ending with the character whose index is *last*. An index of 0 refers to the first character of the **string**. *first* and *last* may be specified as for the **index** method. If *newstring* is specified, then it is placed in the removed character range. If *first* is less than zero then it is treated as if it were zero, and if *last* is greater than or equal to the length of the **string** then it is treated as if it were **end**. If *first* is greater than *last* or the length of the initial **string**, or *last* is less than 0, then the initial **string** is returned untouched.

string **tolower** *string ?first? ?last?*

Returns a value equal to **string** except that all upper (or title) case letters have been converted to lower case. If *first* is specified, it refers to the first char index in the **string** to start modifying. If *last* is specified, it refers to the char index in the **string** to stop at (inclusive). *first* and *last* may be specified as for the **index** method.



string **totitle** *string* *?first?* *?last?*

Returns a value equal to *string* except that the first character in *string* is converted to its Unicode title case variant (or upper case if there is no title case variant) and the rest of the *string* is converted to lower case. If *first* is specified, it refers to the first char index in the *string* to start modifying. If *last* is specified, it refers to the char index in the *string* to stop at (inclusive). *first* and *last* may be specified as for the **index** method.

string **toupper** *string* *?first?* *?last?*

Returns a value equal to *string* except that all lower (or title) case letters have been converted to upper case. If *first* is specified, it refers to the first char index in the *string* to start modifying. If *last* is specified, it refers to the char index in the *string* to stop at (inclusive). *first* and *last* may be specified as for the **index** method.

string **trim** *string* *?chars?*

Returns a value equal to *string* except that any leading or trailing characters from the set given by *chars* are removed. If *chars* is not specified then white space is removed (spaces, tabs, newlines, and carriage returns).

string **trimleft** *string* *?chars?*

Returns a value equal to *string* except that any leading characters from the set given by *chars* are removed. If *chars* is not specified then white space is removed (spaces, tabs, newlines, and carriage returns).

string **trimright** *string* *?chars?*

Returns a value equal to *string* except that any trailing characters from the set given by *chars* are removed. If *chars* is not specified then white space is removed (spaces, tabs, newlines, and carriage returns).

string **wordend** *string* *charIndex*

Returns the index of the character just after the last one in the word containing character *charIndex* of *string*. *charIndex* may be specified as for the **index** method. A word is considered to be any contiguous range of alphanumeric (Unicode letters or decimal digits) or underscore (Unicode connector punctuation) characters, or any single character other than these.

string **wordstart** *string* *charIndex*

Returns the index of the first character in the word containing character *charIndex* of *string*. *charIndex* may be specified as for the **index** method. A word is considered to be any contiguous range of alphanumeric (Unicode letters or decimal digits) or underscore (Unicode connector punctuation) characters, or any single character other than these.



6 Index

3D.....	18	multiple configurations.....	27, 38
areas	55	multiple configurations	16
attribute	50, 51, 52	MyBomTable.....	65
automatisms	59	NLS.....	62
background view	16	numActSheet	56
basics.cfg.....	44	NumSheets	56
BOM	64	OneLevel.....	65
built in values.....	58	PartNumber.....	65
CADSystem	55	pefix - list.....	44
CALL_GUI_Mode.....	44	properties.....	57
callbacks.....	59	revisions.....	21, 45, 48
CATIA	37	screen_height	49, 53
CATIA_TEMPLATES.....	48	screen_width.....	49, 53
change_height.....	49	scroll	55
change_width.....	49	Server.....	35
choosePanelDeltaY	53	SheetScale	56
choosePanelHeight.....	53	showBomMode.....	68
choosePanelWidth.....	53	showUnit_Weight	53
config	20	<i>s/InstallDi</i>	35
configurations.....	38	skin.tcl.....	62
copy	21	sort BOM	68
date.....	24, 54	TD1	37, 60
default	21	<i>TD1_compliant</i>	44
default directory	38	td1_config.cfg.....	60
delete row.....	30	Teilenummer.....	65
Diagrams	15	temporary data sets	68
dp_Weight.....	53	title_block_config.tcl	44
dwgdata_height	49	Titleblock.CATScript.....	14
dwgdata_width.....	49	Trace_Mode	44
enabled	26	u_Units.....	53
English	40	u_Weight.....	53
enhancements	71	units	22
entry_width	53	<i>Units</i>	47, 52, 67
errors	71	Unix	36
extFieldDef.....	50	update 4.0.0.....	15, 16, 38, 40, 44, 47, 48, 50, 60, 68, 71
fields.....	50	update 4.0.2	18, 28, 30, 49
forced	26	update 4.0.3.....	42, 43, 47, 50, 59
global.....	50	update 4.0.4	18, 21, 23, 42, 44, 49, 51, 55, 57, 59
global_fields.....	45	update 4.1.0.....	49, 52, 53, 55, 59, 61
hide bom / show bom	28	update 4.1.1	68
insert row.....	30	update 4.1.2.....	44, 49, 59
language	62	update 4.2.1	52
leadingZeroes	44	update 4.3.0.....	68
mandatory	25	upgrades.....	71
mapper.txt	61	user.....	68
messages.cfg	62	Windows	35
missing	30		