
SMTK Documentation

Release 23.01.100

Kitware, Inc.

Jul 10, 2023

1	CMB User's Guide	3
1.1	Usage	3
1.2	Properties	4
1.3	About Dialog	5
1.4	Default Layout Specification	5
2	Release notes	7
2.1	CMB 23.01 Release Notes	7
2.1.1	Changing to CMB CI Testing	7
2.1.2	SMTK Related Changes	7
2.1.3	ParaView Related Changes	7
2.2	CMB 22.04 Release Notes	7
2.2.1	Simplifying Branding CMB-based Applications	8
2.2.1.1	CMB Refactored to be a template SMTK application	8
2.2.2	UI Improvements	8
2.2.2.1	Animation Features	8
2.2.2.2	Catalyst menu during post-process	8
2.2.2.3	Fixed Postprocessing Mode	8
2.2.3	SMTK Related Changes	8
2.3	CMB 21.12 Release Notes	8
2.3.1	Performance Improvements	8
2.3.1.1	Improving qtAssociation2ColumnWidget's Operation Handler	8
2.3.1.2	Display hint for attribute resources	9
2.3.2	UI Improvements	9
2.3.2.1	Changes in Displaying Attribute Group Items	9
2.3.2.2	Changes in Displaying Attribute Views	9
2.3.2.3	Post-processing control moved to SMTK	10
2.3.3	SMTK Related Changes	10
2.4	CMB 21.07 Release Notes	10
2.4.1	OpenCasacade Support	10
2.4.2	Platform Support	10
2.4.3	SMTK Related Changes	10
3	Indices and tables	11

Version: 23.01.100

Contents:

Computational Model Builder – model builder or CMB for short – is a customizable application to help you (1) describe a simulation's input deck in enough detail that it can be passed to a solver and (2) create input files for a variety of solvers in different simulation packages using your description.

This process can involve any or all of the following:

- importing a geometric model of the simulation domain or the domain's boundary;
- assigning sizing functions to specify mesh element size for analysis;
- submitting portions of the simulation domain to be meshed for analysis;
- assigning material properties to regions of the simulation domain;
- assigning boundary conditions to portions of the simulation domain's boundary;
- assigning initial conditions to portions of the simulation domain or its boundary; and
- assigning global simulation properties such as convergence criteria.

In order to do this, model builder needs a description of your simulation solver. The description is a template which describes the simulation in a JSON or XML document, plus a Python script that examines how the template has been filled out and writes an input file for your simulation. You can provide the template using the editor application that comes with CMB. Once you have the template completed, you can use modelbuilder to fill out the template, associate it with a mesh, and submit it to run.

1.1 Usage

CMB may be configured to use custom branding, such as icons, splash screens, and the about dialog, by setting properties for the project.

By default, the `modelbuilder` application will be built, and the `CMakeLists.txt` and resources to do so can be found under the `modelbuilder/` directory.

1.2 Properties

To set properties for CMB, use the `cmb_set_property` API.

```
cmb_set_property(NAME "MyApplicationName")
```

Properties can be broken into three groups, Application, Resource, and Information.

Table 1.1: Application Properties

Name	Default	Description
NAME	modelbuilder	Exported name of the application.
APP_NAME	<NAME>	Name of the binary executable to build.
VERSION	<PROJECT_VERSION>	Application version.
TITLE	<APP_NAME> <VERSION> <cmb_build_architecture>-bit	Title that is displayed on top of the application window.
ORGANIZATION	Kitware Inc.	Name of the organization that maintains the application.

Application properties are used to configure how different system will display the application in menus:

Table 1.2: Resource Properties

Name	Default	Description
RESOURCE_DIR	modelbuilder/	Change the directory for computing the default resources.
SPLASH	<RESOURCE_DIR>/ <NAME>-splash.png	Splash screen image
ICON	<RESOURCE_DIR>/ <NAME>-icon.png	Icon used for Unix like OSs
WINDOWS_ICON	<RESOURCE_DIR>/win32/ <NAME>.ico	Icon used when building for windows
OSX_ICON	<RESOURCE_DIR>/apple/ <NAME>.icns	Icon used when building for MacOS
LINUX_ICON_SIZES	24x24; 32x32; 96x96	List of icon sizes available for linux desktop
LINUX_ICON	<RESOURCE_DIR>/unix/ <NAME>-<SIZE>.png	Icon associated with the given <SIZE>
APP_DATA	<RESOURCE_DIR>/unix/ <NAME>.appdata.xml	Application information for UNIX app

Resource properties are used to direct the application to image and file resources:

Using the property `RESOURCE_DIR` acts as a replacement for the default search path for resources that are not explicitly specified. The hierarchy for searching for resources is hard-coded and if a resource is not found it will result in an error.

Table 1.3: Information Properties

Name	Default	Description
ORGANIZATION_URL	<code>https://www.kitware.com/</code>	The URL to the organization webpage.
WEBSITE_URL	<code>https://www.computationalmodelbuilder.com</code>	The URL to the application website. (Help->"Website")
USER_GUIDE_URL	<code>https://cmb.readthedocs.io/en/latest/</code>	The URL to the User Guide for this application. (Help->"User's guide and tutorials")
ISSUES_URL	<code>https://gitlab.kitware.com/groups/cmb/-/issues</code>	The URL to the bug reporting page. (Help->"Issue tracker")

Information Properties will show up under the help menu to populate links and/or by the ParaView client when building the application.

1.3 About Dialog

Overriding the help dialog requires a `QObject` class to be defined. A reasonable template to start from can be found in `modelbuilder/mbAboutDialogReaction{.h,.cxx}`. To configure the About dialog use the `cmb_configure_about_dialog` CMake API provided by CMB.

example:

```
cmb_configure_about_dialog(
  CLASS_NAME mbAboutDialogReaction
  SOURCES
    mbAboutDialogReaction.cxx
    mbAboutDialogReaction.h)
```

1.4 Default Layout Specification

Configure the default layout of the Qt dock widgets using the `cmb_configure_layout_spec` CMake API. A reasonable template to start from can be found in `modelbuilder/mbLayoutSpec{.h,.cxx}`.

example:

```
cmb_configure_layout_spec(
  CLASS_NAME mbLayoutSpec
  SOURCES
    mbLayoutSpec.cxx
    mbLayoutSpec.h)
```


2.1 CMB 23.01 Release Notes

See also *CMB 22.04 Release Notes* for previous changes.

CMB 23.01 is based on SMTK 23.01 as well as ParaView 5.11!

2.1.1 Changing to CMB CI Testing

CMB now uses MSVC 2022 to test merge requests rather than MSVC 2019.

2.1.2 SMTK Related Changes

For additional information on changes to SMTK, please see [SMTK-23.01](#).

2.1.3 ParaView Related Changes

For additional information on changes to ParaView, please see [ParaView-5.11](#).

2.2 CMB 22.04 Release Notes

See also *CMB 21.12 Release Notes* for previous changes.

2.2.1 Simplifying Branding CMB-based Applications

2.2.1.1 CMB Refactored to be a template SMTK application

Application details such as icons and naming are now exposed via CMake variables. The `pqAboutDialogReaction` and default `LayoutSpec` may be overridden using application specific implementations.

This will make it easier for developers to create custom CMB-based applications with minimal code duplication.

2.2.2 UI Improvements

2.2.2.1 Animation Features

The ParaView animation features are now included in the postprocessing plugin. When postprocessing is turned on, the Animation View is available in the “View” menu, and a “Save Animation...” item is available in the “File” menu.

2.2.2.2 Catalyst menu during post-process

When post-process mode is active, show the Catalyst menu, and two more entries in the File menu, “Save Catalyst State” and “Save Extracts”

2.2.2.3 Fixed Postprocessing Mode

All of ParaView’s Sources and Filters have been restored when in **postprocessing** mode.

2.2.3 SMTK Related Changes

For additional information on changes to SMTK, please see [SMTK-22.04](#).

2.3 CMB 21.12 Release Notes

See also *CMB 21.07 Release Notes* for previous changes.

2.3.1 Performance Improvements

- Displaying large complex models in the Resource Browser Panel should be much faster.
- Toggling the visibility of a large number of model components in the Resource Browser should be much faster.

2.3.1.1 Improving `qtAssociation2ColumnWidget`’s Operation Handler

The original code would cause a large number of refreshes due to the fact it merely would see if there were any resource changes. The new logic is a bit more intelligent and will only cause a refresh if any of the following conditions happen:

- A modified component is either the widget’s attribute or can be associated with the widget’s attribute
- A deleted component could have been associated with the widget’s attribute
- A created component could be associated with the widget’s attribute

The result should be a more responsive UI.

2.3.1.2 Display hint for attribute resources

CMB ModelBuilder now supports version 5 SMTK XML files that include the “DisplayHint” XML attribute for Attribute Resources. Only those Attribute Resource with this hint will be automatically displayed in the attribute editor panel when loaded. Older XML files will still automatically display themselves in the attribute editor panel when loaded. This can result in speed improvements in CMB applications that load in multiple SBT files at the same time.

2.3.2 UI Improvements

2.3.2.1 Changes in Displaying Attribute Group Items

Empty Group Items will no longer display an empty frame.

2.3.2.2 Changes in Displaying Attribute Views

Hiding “Advance Level” Definitions

Attribute Views will now hide any attribute definitions that have an advance level that is higher than the user’s active advance level. This enables the Attribute View to hide itself if all its definitions should be hidden from the user.

Improving UI handling of Signal Operations

Originally the `qtAttributeView` class would ignore the Signal Operation since typically it would be the only Qt UI element that would be creating, removing, and changing the Attributes it is displaying. However, this prevented the UI designer from having AttributeViews that displayed the same information from being used in Selector Views or have different AttributeViews overlap their contents (for example one View could be displaying Fluid Boundary Conditions, while another was displaying all Boundary Conditions)

This change now encodes the address of the View that initiated the change so that we can avoid a View from being updated via a Signal Operation that it itself initiated.

`qtAttributeView` has now been updated to only ignore Signal Operations that it triggered.

Supporting `smtk.extensions.attribute_view.name_read_only` in `qtAttributeViews`

You can now indicate that an Attribute’s name should not be modified by creating a bool Property on the Attribute called: `smtk.extensions.attribute_view.name_read_only` and setting its value to true.

Added Ability to Set Attribute Editor Panel’s Title

The Attribute Editor Panel name can now be configure by a `smtk::view::Configuration`.

If the Configuration is Top Level then the following Configuration Attributes can be used:

- `AttributePanelTitle` - defines the base name of the Panel. If not specified it defaults to Attribute Editor.
- `IncludeResourceNameInPanel` - if specified and set to true, the Panel’s title will include the name of the resource in ()

`SimpleAttribute.sbt` contains an example:

```
<Views>
  <View Type="Attribute" Title="External Expression Test Pt - Source" TopLevel="true"
  ↳DisableTopButtons="false"
    AttributePanelTitle="SMTK Test" IncludeResourceNameInPanel="t">
      <AttributeTypes>
        <Att Type="B-expressions"/>
      </AttributeTypes>
    </View>
</Views>
```

2.3.2.3 Post-processing control moved to SMTK

In order for SMTK's attribute panel to modify its behavior depending on whether CMB is in post-processing mode or not, the state variable that controls post-processing mode has been moved into SMTK's `pqSMTKBehavior` class.

The toolbar button and UI code to effect post-processing mode remain in this repository (CMB). However, now it is possible for other SMTK elements to initiate post-processing mode.

2.3.3 SMTK Related Changes

For additional information on changes to SMTK, please see [SMTK-21.12](<https://smtk.readthedocs.io/en/latest/release/smtk-21.12.html>)

2.4 CMB 21.07 Release Notes

See also [CMB 21.05 Release Notes](#) for previous changes.

2.4.1 OpenCasacade Support

Modelbuilder now comes with support to load IGES, STEP, and OpenCascade files via [OpenCascade](#) . Only viewing is currently supported. In the future, annotations on CAD models as well as modeling operations will be added.

In addition, OpenCascade SMTK Resources can now be saved and loaded into ModelBuilder.

Finally, distributed binaries will now be built with OpenCascade support.

2.4.2 Platform Support

ARM-Macs are now supported.

2.4.3 SMTK Related Changes

For additional information on changes to SMTK, please see [SMTK-21.07](<https://gitlab.kitware.com/cmb/smtk/-/blob/master/doc/release/smtk-21.07.rst>)

CHAPTER 3

Indices and tables

- `genindex`
- `modindex`
- `search`