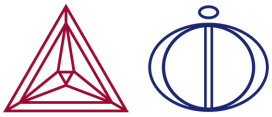
TC-Toolbox for MATLAB®

SDK Programmer's Guide

Thermo-Calc Version 2020b



Thermo-Calc Software

Copyright 2020 Thermo-Calc Software AB. All rights reserved.

Information in this document is subject to change without notice. The software or database described in this document is furnished under a license agreement or nondisclosure agreement. The software or database may be used or copied only in accordance with the terms of those agreements.

Thermo-Calc Software AB

Råsundavägen 18, SE-169 67 Solna, Sweden

+46 8 545 959 30

documentation@thermocalc.com

www.thermocalc.com

Contents

Introduction to the TC-Toolbox for MATLAB®

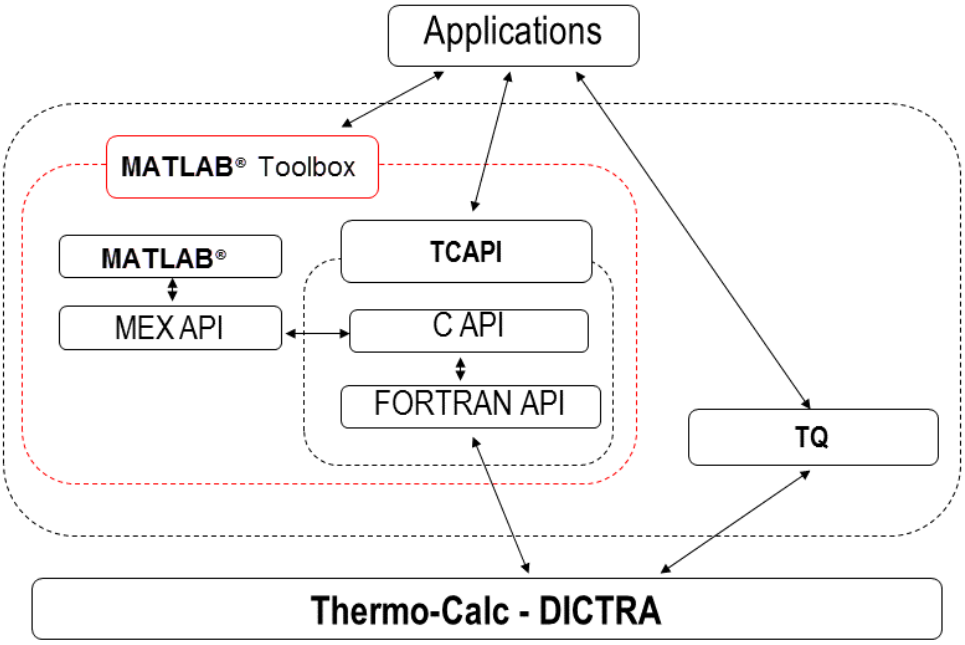
Thermo-Calc is a general software package for manipulation of thermodynamic quantities and multicomponent phase equilibrium calculations. Currently, there are three application programming interfaces available for Thermo-Calc: TQ-Interface, TC-API and TC-Toolbox for MATLAB. In this guide TC-Toolbox for MATLAB, the interface between Thermo-Calc and MATLAB® is discussed.

In this section:

About TC-Toolbox for MATLAB®

The concept of the application programming interfaces for Thermo-Calc is that an application programmer does not need to understand the Thermo-Calc kernel but can use its powerful features in other programs.

MATLAB® is a flexible software for technical computing and visualization of data. The software has more than 600 mathematical, statistical and engineering functions, and graphical capabilities. It is a matrix-oriented programming language and contains compilers, links and libraries for different scientific applications. This flexibility is enhanced with being able to retrieve thermodynamic and kinetic quantities through the TC-Toolbox for MATLAB®. This programming interface is ideal for fast realization of ideas during research and development activities.



To be able to call MATLAB from programs written in C or FORTRAN there are MEX-files (MATLAB Executable) included with the MATLAB software. These MEX-files were utilised when interfacing MATLAB with Thermo-Calc.

For every Thermo-Calc function implemented in the MEX-files there is a corresponding m-file, making it possible to call Thermo-Calc from MATLAB just by running the corresponding m-file.

More than 50 commands are available for the application programmer. For more information, general functionality and applications of the MATLAB software refer to the documentation provided by the MathWorks Ltd. (www.mathworks.com/help/).

Installing TC-Toolbox for MATLAB®

TC-Toolbox for MATLAB® needs to be installed on the same computer or on a server with the Thermo-Calc software and database package. TC-Toolbox for MATLAB is available for Windows operating systems.

For installation details, see in the *Thermo-Calc Installation Guide* .

Test the Installation

Once the installation is complete, you can test the connectivity in MATLAB®.

Start MATLAB and type: **tc_init_root** in the command window and press return. This should result in no return message for a successful installation. All of the commands available in the toolbox are described in this document.

To get a short description of each command type in the command window **help Thermo-Calc-Toolbox** *X* (where *X* is the installed version number of the toolbox).

TC-Toolbox for MATLAB Examples

Examples for the TC-Toolbox are placed under MATLAB® in the same documents folder as the Thermo-Calc files (Documents folder):

```
..\Documents\MATLAB\Thermo-Calc-Toolbox-X\Examples
```

Where *X* is the installed version number of the toolbox.

For most installations the examples are available in the MATLAB window when the software is opened.

For installation details, see in the *Thermo-Calc Installation Guide*.

Example Descriptions

Example Name	Description
ex01.m	Calculation of a single equilibrium in Fe-Cr-C at 1200 K.
ex02.m	Calculation of a molar Gibbs energy surface in an Al-Cu-Si alloy.
calc_para_eq.m	Example 3. calc_para_eq.m (calls the functions paraf.m and qparaf.m) Calculation of paraequilibrium and quasi-paraconditions for an alloy with at least one interstitial

Example Name	Description
paraf.m	component (e.g. N or C).
qparaf.m	Demonstrates also the coding of an interactive program.
ex04.m	Calculation of the so-called T-zero line in Fe-C.
ex05.m	Calculation of the influence of composition on the A3 temperature in an Fe-Cr-C alloy. The A3 temperature is calculated for a large number of uniformly distributed compositions in composition space. The carbon content belongs to the interval [1E-4:5E-3] (weight fraction) and the cromium content belongs to the interval [1E-2,3E-2]. The relative frequency (the fraction) of compositions belonging to a certain A3 temperature interval is then plotted.
ex06_ interfacial_ energy.m	Calculation of interfacial energy between BCC and M7C3 for a Fe-12Cr-0.1C alloy.

Commands in TC-Toolbox for MATLAB®

To avoid conflict with reserved names all commands in the TC-Toolbox for MATLAB® start with **tc_** and the DICTRA module commands start with **dic_**.

Group	Description
tc_root	General information and miscellaneous commands
tc_database	Information and commands in the database module
tc_system	Information and commands in the database module
tc_util	Various commands e.g. “tc_define_system”
tc_ges5	Information and commands in the GES5 module
dic_dictra	Information and commands in the DICTRA module

tc_root

Name	Arguments	Description
`tc_init_root`	None	Initialise the Thermo-Calc subsystem. Must be called before any other command in the Toolbox.
`tc_deinit`	None	Closes the Thermo-Calc session and returns the license key.
`tc_version`	string: version_name	Returns the current version of the Thermo-Calc subsystem.
`tc_poly3_command`	string: command	Sends a command to the POLY-3 module.
`tc_read_poly3_file`	string: file_name	Reads stored POLY-3 file name.
`tc_save_poly3_file`	string: file_name	Saves a POLY-3 file name.

tc_database

Name	Arguments	Description
`tc_append_database`	string: database_name	Appends database name.
`tc_element_select`	string: element_name	Selects an element name from the current database.
`tc_get_data`	None	Executes the GET_DATA command.
`tc_open_database`	string: database_name	Opens a named database.

Name	Arguments	Description
`tc_phase`	integer: no_phases string array phase_names	Returns the number (no.) of phases and the phase names.
`tc_phase_reject`	string: phase_name	Rejects phase name in the current database.
`tc_phase_select`	string: phase_name	Selects phase name in the current database.

tc_system

Name	Arguments	Description
`tc_error`	integer: error_code string: error_message	Checks if an error occurred, then returns an error code and message.
`tc_reset_error`	None	Resets the error handling in the Thermo-Calc subsystem
`tc_compute_equilibrium`	None	Executes the COMPUTE_EQULIBRIUM command in POLY-3
`tc_component_status`	string: status string: comp_name	Returns the status for component (comp) name. The status can be ENTERED or SUSPENDED.
`tc_create_new_equilibrium`	integer: eq_number	Create a new equilibrium with equilibrium number.
`tc_define_components`	string: new_components	Changes the set of components to those in new components.
`tc_degrees_of_freedom`	integer: number	Returns the degrees of freedom number in the system.
`tc_delete_condition`	string: condition_name	Deletes the named condition.
`tc_delete_symbol`	string: symbol_name	Deletes the named symbol.
`tc_enter_constant`	string: constant_name double: value	Enters a symbol of type CONSTANT with constant_name and value.
`tc_enter_function`	string: function_name string: function_expression	Enters a symbol of type FUNCTION with function_name and expression.
`tc_enter_symbol`	string: symbol_name string: symbol_type integer: argument_type	Enters a named symbol and type (=CONSTANT, FUNCTION, TABLE or VARIABLE) with an argument type (=1 for integer, 2 for double or 3 for string).

Name	Arguments	Description
	integer: int_value double: double_value string: char_value	
`tc_enter_table`	string: table_name string: table_expression	Enters a symbol of type TABLE with table_ name and expression.
`tc_enter_variable`	string: variable_name double: value	Enters a symbol of type VARIABLE with variable_name and value.
`tc_get_derivatives`	string: phase string array: arr1 string array: arr2	Returns the Gibbs energy and the first and second derivatives with respect to site-fractions for phase. The array arr1 contains the Gibbs energy and the first derivatives and the array arr2 contains the second derivatives.
`tc_get_surface_energy`	Input parameters: string with matrix phase name string with precipitate phase name integer index of dependent component in u-fractions (where component list is sorted alphabetically) double with the temperature array of doubles containing the u-fractions (where component list first is sorted alphabetically) double with the molar volume of the matrix phase double with the molar volume of the precipitating phase	 The list of components should be sorted alphabetically and special components (e.g VA) removed. See the*ex06_ interfacial_energy.m* example for more detail. Retrieves the surface energy with unit J/m^2
`tc_get_value`	string: expression double: value	Retrieves the current value of any state variable, function or variable set in expression.
`tc_list_component`	integer: no_components string array: components	Returns the number (no.) of components and a list of all components.

Name	Arguments	Description
`tc_list_conditions`	integer: no_conditions string array: conditions	Returns the number (no.) of conditions and a list of all conditions.
`tc_list_phase`	integer: no_phases string array: phases	Returns the number (no.) of phases and a list of all phases.
`tc_list_species`	integer: no_species string array: species	Returns the number (no.) of species and a list of all species.
`tc_list_symbols`	integer: no_symbols string array: symbols	Returns the number (no.) of symbols and a list of all defined symbols.
`tc_phase_status`	string: status string: phase_name	Returns the status for the named phase.
`tc_select_equilibrium`	integer: eq_number	Command to switch to another set of conditions and equilibria. The desired set of conditions and equilibria are indicated by its equilibrium (eq) number.
`tc_set_component_status`	string: comp_name string: status	Sets the status (ENTERED or SUSPENDED) for a named component.
`tc_set_condition`	string: expression double: value	Sets a condition for expression to value.
`tc_set_minimization`	string: flag	Turns global minimization on or off by setting the string flag to on or off.
`tc_set_phase_addition`	string: phase_name double: value	Command to add a value to the Gibbs energy expression of a named phase.
`tc_set_phase_status`	string: phase_name string: status double: value	Sets status (ENTERED, DORMANT, FIXED or SUSPENDED) to a named phase. A value is to set for status ENTERED and FIXED.
`tc_set_start_value`	string: name double: value	Sets a start value for a state variable name.
`tc_species_status`	string: status string: species_name	Returns the status for a named species.

tc_util

Name	Arguments	Description
`tc_check_error`	string:	Checks for errors and resets them. It is a combination of tc_error and tc_reset_error.
`tc_define_system`	string: database_name string: element_names string: reject_phases string: restore_phases	Define a system with a named database, element names, phases to reject and phases to restore.
`tc_prompt`	string: tprompt integer: defval	Prompt to input an integer value.
`tc_promptr`	string: tprompt double:defval	Prompt to input a double value.
`tc_prompts`	string: tprompt string: defval	Prompt to input a string.
`tc_promptsn`	string: tprompt string array: defval	Prompt to input a string array.

tc_ges5

Name	Arguments	Description
`tc_enter_ges5_parameter`	string: parameter_name string: parameter_expression	Enters a named parameter in parameter_expression.
`tc_ges5_command`	string: command	Sends a command to the GES5 monitor.
`tc_get_ges5_parameter`	string: parameter_expression string: parameter_name	Returns a parameter_expression for parameter_name.

dic_dictra

A Diffusion Module (DICTRA) license is required to use these commands.

Name	Arguments	Description
`dic_command`	string: command	Sends a command to the DICTRA module.
`dic_convert_sitefractions`	double array: new_fractions string: phase_name double array: sitefractions integer: fraction_type	Convert site fractions in for a named phase. Set new fractions and fraction type=1, 2, 3 return mole-, mass- or u-fractions, respectively.
`dic_get_independent_component`	integer: no_idepc string array: comp_names string region_name	Returns the number (no.) of independent components (idepc) and a list of component names for a named region.
`dic_list_profile`	integer: no_gridpoints integer: no_sitefractions double: sitefractions double array: gridpoints string: region_name string: phase_name	Returns a stored profile for a named phase and region.
`dic_list_timesteps`	integer: no_timesteps double array: timesteps	Returns the number (no.) of time steps and a list of time steps.
`dic_read_workspace`	string: file_name	Reads the stored simulation file name.
`dic_region_info`	integer: no_gridpoints double: region_size double: start_coordinate string: region_name	Returns information about the named region: the size of the region, number (no.) of grid points and value of the first (start) coordinate.
`dic_save_workspace`	string: file_name	Saves a simulation file to a new name.
`dic_select_timestep`	integer: time_step	Selects a time step from a stored simulation file.
`dic_simulate_reaction`	None	Start the simulation.