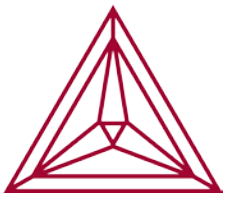
TC-Toolbox for MATLAB®

SDK Programmer's Guide for Thermo-Calc Version 2015a



TC-Toolbox for MATLAB®
SDK Programmer's Guide

Contents

1. Introduction

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® (see Figure 1). In this guide TC-Toolbox for MATLAB®, the interface between Thermo-Calc and MATLAB®, is discussed.

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.

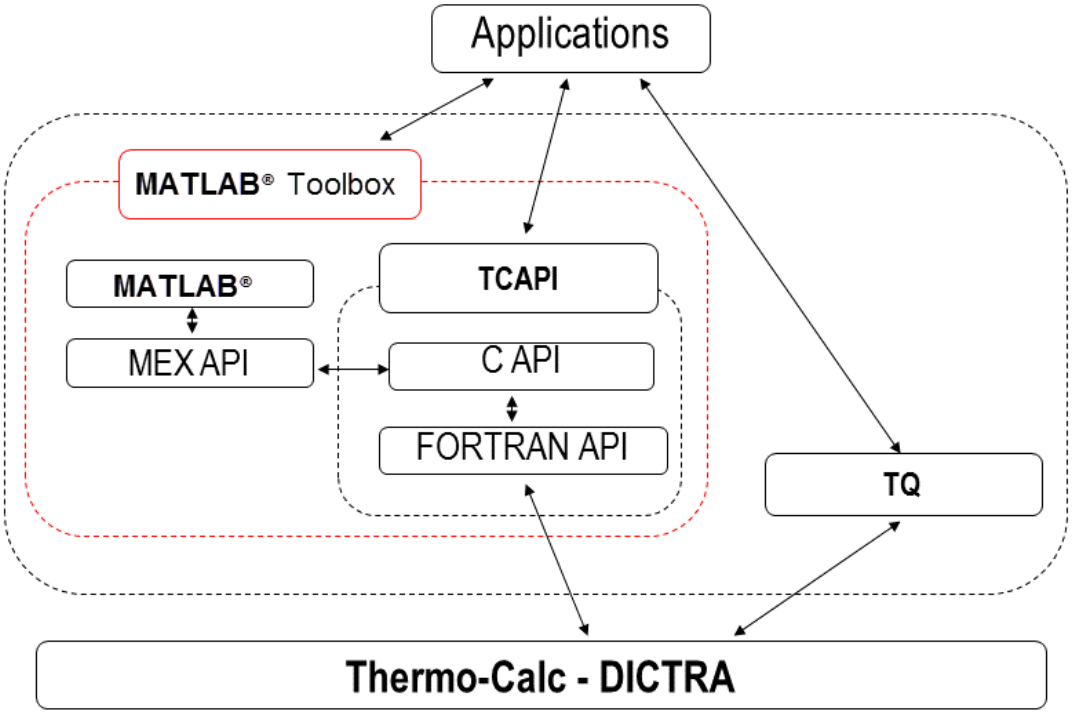


Figure 1 Example of the available APIs for Thermo-Calc.

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 (Figure 1).

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.

In the current version of the TC-Toolbox for MATLAB® 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/).

2. Installing and running 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 the Thermo-Calc Installation Guide for more information (http://www.thermocalc.com/support/documentation/).

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 MATLAB). Examples can be found in the Thermo-Calc toolbox folder.

3. Commands in TC-Toolbox for MATLAB®

The commands in the TC-Toolbox for MATLAB® are grouped by purpose:

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 DICTRA


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

3.1 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 `file_name`.
`tc_save_poly3_file`	string: file_name	Saves a POLY-3 file in `file_name`.

3.2 TC_DATABASE

Name	Arguments	Description
`tc_append_database`	string: database_name	Appends database `database_name`.
`tc_element_select`	string: element_name	Selects an element `element_name` from the current database.
`tc_get_data`	None	Executes the `GET_DATA` command.
`tc_open_database`	string: database_name	Opens database `database_name`.
`tc_phase`	integer: no_phases string array phase_names	Returns the number of phases in `no_phases` and `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.

3.3 TC_SYSTEM

Name	Arguments	Description
`tc_error`	integer: error_code string: error_message	Checks if an error occurred. If so an `error_code` and an `error_message` are returned.
`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`, `status` can be `ENTERED` or `SUSPENDED`.
`tc_create_new_equilibrium`	integer: eq_number	Command to create a new equilibrium with equilibrium number `eq_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 condition `condition_name`.
`tc_delete_symbol`	string: symbol_name	Deletes the `symbol_name`.
`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 integer: int_value double: double_value string: char_value	Enters a `symbol_name` of type `symbol_type` (=`CONSTANT`, `FUNCTION`, `TABLE` or `VARIABLE`) with an `argument_type` (=1 for integer, 2 for double or 3 for string).

Name	Arguments	Description
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_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 of components in no_components and a list of all components.
tc_list_conditions	integer: no_conditions string array: conditions	Returns the number of conditions in no_conditions and a list of all conditions in conditions.
tc_list_phase	integer: no_phases string array: phases	Returns the number of phases in no_phases and a list of all phases.
tc_list_species	integer: no_species string array: species	Returns the number of species in no_species and a list of all species.
tc_list_symbols	integer: no_symbols string array: symbols	Returns the number of symbols in no_symbols and a list of all defined symbols.
tc_phase_status	string: status string: phase_name	Returns the status for the phase in phase_name.
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 number eq_number.

Name	Arguments	Description
tc_set_component_status	string: comp_name string: status	Sets the status (ENTERED or SUSPENDED) for component comp_name.
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 phase phase_name.
tc_set_phase_status	string: phase_name string: status double: value	Sets status (ENTERED, DORMANT, FIXED or SUSPENDED) to phase phase_name. 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 species species_name.

3.4 TC_UTIL

Name	Arguments	Description
tc_check_error	string:	Checks for errors and resets them. This command 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	A single command to define a system with database database_name, elements element_names, phases to reject in reject_phases and phases to restore in restore_phases.
tc_prompt	string: tprompt integer: defval	Prompt to input a integer value.
tc_promptr	string: tprompt double:defval	Prompt to input a double value.

Name	Arguments	Description
tc_prompts	string: tprompt string: defval	Prompt to input a string.
tc_promptsn	string: tprompt string array: defval	Prompt to input a string array.

3.5 TC_GES5

Name	Arguments	Description
tc_enter_ges5_parameter	string: parameter_name string: parameter_expression	Enters a parameter parameter_name 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.

3.6 DIC_DICTRA

Name	Arguments	Description
dic_command	string: command	Sends a command to the DICTRA monitor.
dic_convert_sitefractions	double array: new_fractions string: phase_name double array: sitefractions integer: fraction_type	Convert site fractions in sitefractions for phase phase_name. New fractions is set in new_fractions. 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 of independent components in no_idepc and a list of component names in comp_names for region region_name.
dic_list_profile	integer: no_gridpoints	Returns a stored profile for phase_name and

Name	Arguments	Description
	integer: no_sitefractions double: sitefractions double array: gridpoints string: region_name string: phase_name	`region_name`.
`dic_list_timesteps`	integer: no_timesteps double array: timesteps	Returns the number of time steps in `no_timesteps` and a list of time steps in `timesteps`.
`dic_read_workspace`	string: file_name	Reads the stored DICTRA simulation file in `file_name`.
`dic_region_info`	integer: no_gridpoints double: region_size double: start_coordinate string: region_name	Returns information about `region_name`: the size of the region in `region_size`, number of grid points in `no_gridpoints` and value of the first coordinate in `start_coordinate`.
`dic_save_workspace`	string: file_name	Saves a DICTRA simulation file to `file_name`.
`dic_select_timestep`	integer: time_step	Selects a time step from a stored DICTRA simulation file.
`dic_simulate_reaction`	None	Command to start the simulation.