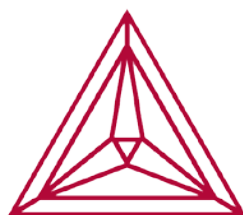


# **TC-API: SDK Programmer's Guide**

**Thermo-Calc Version 2018b**



Copyright 2018 Thermo-Calc Software AB. All rights reserved.

Information in this document is subject to change without notice. The software described in this document is furnished under a license agreement or nondisclosure agreement. The software 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](mailto:documentation@thermocalc.com)

[www.thermocalc.com](http://www.thermocalc.com)

Copyright 2018 Thermo-Calc Software AB. All rights reserved.

Information in this document is subject to change without notice. The software described in this document is furnished under a license agreement or nondisclosure agreement. The software 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](mailto:documentation@thermocalc.com)

[www.thermocalc.com](http://www.thermocalc.com)

TC-API  
7.8.11410

Generated by Doxygen 1.8.9.1

Fri Sep 29 2017 19:50:12



# Contents

<b>1</b>	<b>Thermo-Calc c-api</b>	<b>1</b>
1.1	Installed files . . . . .	1
1.1.1	TC-API libraries. . . . .	1
1.1.2	Source folder. . . . .	1
1.1.3	Project folders for building the example code . . . . .	1
1.2	Explicit loading or Dynamic linking . . . . .	1
1.3	About the source code . . . . .	2
1.3.1	The dynamic linking examples use the files . . . . .	2
1.3.2	The explicit loading examples use the files . . . . .	2
1.4	When starting developing proprietary projects the following code is needed . . . . .	2
1.4.1	The dynamic linking examples use the files . . . . .	2
1.4.2	The explicit loading examples use the files . . . . .	2
<b>2</b>	<b>Class Index</b>	<b>5</b>
2.1	Class List . . . . .	5
<b>3</b>	<b>File Index</b>	<b>7</b>
3.1	File List . . . . .	7
<b>4</b>	<b>Class Documentation</b>	<b>9</b>
4.1	<code>_tc_function_library</code> Struct Reference . . . . .	9
4.1.1	Detailed Description . . . . .	10
4.1.2	Member Data Documentation . . . . .	10
4.1.2.1	<code>tc_append_database</code> . . . . .	10
4.1.2.2	<code>tc_check_license</code> . . . . .	10
4.1.2.3	<code>tc_component_status</code> . . . . .	10
4.1.2.4	<code>tc_compute_equilibrium</code> . . . . .	10
4.1.2.5	<code>tc_create_new_equilibrium</code> . . . . .	10
4.1.2.6	<code>tc_database</code> . . . . .	11
4.1.2.7	<code>tc_define_components</code> . . . . .	11
4.1.2.8	<code>tc_degrees_of_freedom</code> . . . . .	11
4.1.2.9	<code>tc_deinit</code> . . . . .	11

4.1.2.10	tc_delete_condition	11
4.1.2.11	tc_delete_symbol	11
4.1.2.12	tc_element	11
4.1.2.13	tc_element_reject	11
4.1.2.14	tc_element_select	11
4.1.2.15	tc_enter_ges5_parameter	11
4.1.2.16	tc_enter_symbol	11
4.1.2.17	tc_error	11
4.1.2.18	tc_ges5	12
4.1.2.19	tc_get_data	12
4.1.2.20	tc_get_derivatives	12
4.1.2.21	tc_get_ges5_parameter	12
4.1.2.22	tc_get_value	12
4.1.2.23	tc_init_root	12
4.1.2.24	tc_init_root3	12
4.1.2.25	tc_list_component	12
4.1.2.26	tc_list_conditions	12
4.1.2.27	tc_list_phase	12
4.1.2.28	tc_list_species	12
4.1.2.29	tc_list_symbols	12
4.1.2.30	tc_nr_of_constituents_in_phase	13
4.1.2.31	tc_open_database	13
4.1.2.32	tc_phase	13
4.1.2.33	tc_phase_all_constituents	13
4.1.2.34	tc_phase_constituents	13
4.1.2.35	tc_phase_reject	13
4.1.2.36	tc_phase_select	13
4.1.2.37	tc_phase_status	13
4.1.2.38	tc_phase_structure	13
4.1.2.39	tc_poly3	13
4.1.2.40	tc_put_sitefractions	13
4.1.2.41	tc_read_poly3_file	13
4.1.2.42	tc_reject_constituent	14
4.1.2.43	tc_reset_error	14
4.1.2.44	tc_restore_constituent	14
4.1.2.45	tc_save_poly3_file	14
4.1.2.46	tc_select_equilibrium	14
4.1.2.47	tc_set_component_status	14
4.1.2.48	tc_set_condition	14
4.1.2.49	tc_set_license_code	14

4.1.2.50	tc_set_minimization_option	14
4.1.2.51	tc_set_phase_addition	14
4.1.2.52	tc_set_phase_status	14
4.1.2.53	tc_set_start_value	14
4.1.2.54	tc_species_status	15
4.1.2.55	tc_version	15
4.2	tc_components_strings Struct Reference	15
4.2.1	Detailed Description	15
4.2.2	Member Data Documentation	15
4.2.2.1	component	15
4.3	tc_conditions_as_arrays_of_strings Struct Reference	15
4.3.1	Detailed Description	15
4.3.2	Member Data Documentation	16
4.3.2.1	condition	16
4.4	tc_constituents_strings Struct Reference	16
4.4.1	Detailed Description	16
4.4.2	Member Data Documentation	16
4.4.2.1	constituent	16
4.5	tc_databases_strings Struct Reference	16
4.5.1	Detailed Description	16
4.5.2	Member Data Documentation	16
4.5.2.1	database	16
4.6	tc_elements_strings Struct Reference	17
4.6.1	Detailed Description	17
4.6.2	Member Data Documentation	17
4.6.2.1	element	17
4.7	tc_phases_strings Struct Reference	17
4.7.1	Detailed Description	17
4.7.2	Member Data Documentation	17
4.7.2.1	phase	17
4.8	tc_reference_strings Struct Reference	17
4.8.1	Detailed Description	18
4.8.2	Member Data Documentation	18
4.8.2.1	reference	18
4.9	tc_species_strings Struct Reference	18
4.9.1	Detailed Description	18
4.9.2	Member Data Documentation	18
4.9.2.1	specie	18

## 5 File Documentation

19



5.1	example1.c File Reference	19
5.2	example2.c File Reference	19
5.2.1	Function Documentation	19
5.2.1.1	main	19
5.3	example3.c File Reference	19
5.3.1	Typedef Documentation	20
5.3.1.1	ivect	20
5.3.1.2	rvect	20
5.3.1.3	str8	20
5.3.1.4	strvect	20
5.3.2	Function Documentation	20
5.3.2.1	main	20
5.4	libtc.c File Reference	20
5.4.1	Function Documentation	20
5.4.1.1	importFunctions	20
5.4.1.2	tloadfunc	21
5.5	libtc.h File Reference	21
5.5.1	Typedef Documentation	22
5.5.1.1	BoolFuncIntPStringInt	22
5.5.1.2	BoolFuncStringStringStrLen	22
5.5.1.3	FloatFuncString	22
5.5.1.4	IntFuncNoParams	22
5.5.1.5	IntFuncString	22
5.5.1.6	IntFuncStringInt	22
5.5.1.7	IntFuncStringIntIntP	22
5.5.1.8	IntFuncStringIntPStringIntFloatP	22
5.5.1.9	IntFuncStringIntPStringStrLenFloatP	22
5.5.1.10	IntFuncStringString	22
5.5.1.11	StringFuncString	22
5.5.1.12	tc_function_library	23
5.5.1.13	VoidFuncInt	23
5.5.1.14	VoidFuncIntPIntPIntPIntP	23
5.5.1.15	VoidFuncNoParams	23
5.5.1.16	VoidFuncString	23
5.5.1.17	VoidFuncStringFloat	23
5.5.1.18	VoidFuncStringFloatP	23
5.5.1.19	VoidFuncStringFloatPFloatP	23
5.5.1.20	VoidFuncStringInt	23
5.5.1.21	VoidFuncStringIntInt	23
5.5.1.22	VoidFuncStringIntString	23

5.5.1.23	VoidFuncStringString	23
5.5.1.24	VoidFuncStringStringFloat	23
5.5.1.25	VoidFuncStringStringInt	24
5.5.1.26	VoidFuncStringStringIntIntFloatString	24
5.5.2	Function Documentation	24
5.5.2.1	importFunctions	24
5.6	ReadMe.txt File Reference	24
5.7	ReadMe.txt File Reference	24
5.8	tc_data_defs.h File Reference	24
5.8.1	Macro Definition Documentation	25
5.8.1.1	BOOL_FUNC_WIN	25
5.8.1.2	DllExport	25
5.8.1.3	false	25
5.8.1.4	FLOAT_FUNC_WIN	26
5.8.1.5	INTEGER_FUNC	26
5.8.1.6	INTEGER_FUNC_GNU	26
5.8.1.7	INTEGER_FUNC_WIN	26
5.8.1.8	TC_EPS	26
5.8.1.9	TC_MAX_NR_OF_AXES	26
5.8.1.10	TC_MAX_NR_OF_CONST_PER_SUBLATTICE	26
5.8.1.11	TC_MAX_NR_OF_CONST_PER_SUBLATTICE_IN_IDEAL_GAS	26
5.8.1.12	TC_MAX_NR_OF_CONSTITUENTS	26
5.8.1.13	TC_MAX_NR_OF_DATABASES	26
5.8.1.14	TC_MAX_NR_OF_ELEMENTS	26
5.8.1.15	TC_MAX_NR_OF_PHASES	26
5.8.1.16	TC_MAX_NR_OF_SPECIES	27
5.8.1.17	TC_MAX_NR_OF_SUBLATTICES	27
5.8.1.18	TC_NWSE	27
5.8.1.19	TC_NWSG	27
5.8.1.20	TC_STRLEN_COMPONENTS	27
5.8.1.21	TC_STRLEN_CONSTITUENTS	27
5.8.1.22	TC_STRLEN_DATABASE	27
5.8.1.23	TC_STRLEN_ELEMENTS	27
5.8.1.24	TC_STRLEN_MAX	27
5.8.1.25	TC_STRLEN_PATH_MAX	27
5.8.1.26	TC_STRLEN_PHASES	27
5.8.1.27	TC_STRLEN_REFERENCE	27
5.8.1.28	TC_STRLEN_SPECIES	28
5.8.1.29	TC_STRLEN_STOICHIOMETRY	28
5.8.1.30	TC_VARS	28

5.8.1.31	TCFuncExport	28
5.8.1.32	TCHANDLE	28
5.8.1.33	true	28
5.8.1.34	VOID_FUNC_WIN	28
5.8.2	Typedef Documentation	28
5.8.2.1	pointer	28
5.8.2.2	TC_BOOL	28
5.8.2.3	tc_components_strings	28
5.8.2.4	tc_conditions_as_arrays_of_strings	28
5.8.2.5	tc_constituents_strings	28
5.8.2.6	tc_databases_strings	29
5.8.2.7	tc_elements_strings	29
5.8.2.8	TC_FLOAT	29
5.8.2.9	TC_IARR	29
5.8.2.10	TC_INT	29
5.8.2.11	TC_LABEL_STRING	29
5.8.2.12	tc_phases_strings	29
5.8.2.13	tc_reference_strings	29
5.8.2.14	tc_species_strings	29
5.8.2.15	TC_STRING	29
5.8.2.16	TC_STRING_LENGTH	29
5.9	tcapi.h File Reference	29
5.9.1	Function Documentation	31
5.9.1.1	tc_append_database	31
5.9.1.2	tc_check_license	31
5.9.1.3	tc_component_status	31
5.9.1.4	tc_compute_equilibrium	31
5.9.1.5	tc_create_new_equilibrium	31
5.9.1.6	tc_database	31
5.9.1.7	tc_define_components	31
5.9.1.8	tc_degrees_of_freedom	31
5.9.1.9	tc_deinit	31
5.9.1.10	tc_delete_condition	31
5.9.1.11	tc_delete_symbol	32
5.9.1.12	tc_element	32
5.9.1.13	tc_element_reject	32
5.9.1.14	tc_element_select	32
5.9.1.15	tc_enter_ges5_parameter	32
5.9.1.16	tc_enter_symbol	32
5.9.1.17	tc_error	32

5.9.1.18	<code>tc_ges5</code>	32
5.9.1.19	<code>tc_get_data</code>	32
5.9.1.20	<code>tc_get_derivatives</code>	32
5.9.1.21	<code>tc_get_ges5_parameter</code>	32
5.9.1.22	<code>tc_get_value</code>	33
5.9.1.23	<code>tc_init_root</code>	33
5.9.1.24	<code>tc_init_root3</code>	33
5.9.1.25	<code>tc_list_component</code>	33
5.9.1.26	<code>tc_list_conditions</code>	33
5.9.1.27	<code>tc_list_phase</code>	33
5.9.1.28	<code>tc_list_species</code>	33
5.9.1.29	<code>tc_list_symbols</code>	33
5.9.1.30	<code>tc_nr_of_constituents_in_phase</code>	33
5.9.1.31	<code>tc_open_database</code>	33
5.9.1.32	<code>tc_phase</code>	33
5.9.1.33	<code>tc_phase_all_constituents</code>	34
5.9.1.34	<code>tc_phase_constituents</code>	34
5.9.1.35	<code>tc_phase_reject</code>	34
5.9.1.36	<code>tc_phase_select</code>	34
5.9.1.37	<code>tc_phase_status</code>	34
5.9.1.38	<code>tc_phase_structure</code>	34
5.9.1.39	<code>tc_poly3</code>	34
5.9.1.40	<code>tc_put_sitefractions</code>	34
5.9.1.41	<code>tc_read_poly3_file</code>	34
5.9.1.42	<code>tc_reject_constituent</code>	34
5.9.1.43	<code>tc_reset_error</code>	34
5.9.1.44	<code>tc_restore_constituent</code>	35
5.9.1.45	<code>tc_save_poly3_file</code>	35
5.9.1.46	<code>tc_select_equilibrium</code>	35
5.9.1.47	<code>tc_set_component_status</code>	35
5.9.1.48	<code>tc_set_condition</code>	35
5.9.1.49	<code>tc_set_license_code</code>	35
5.9.1.50	<code>tc_set_minimization_option</code>	35
5.9.1.51	<code>tc_set_phase_addition</code>	35
5.9.1.52	<code>tc_set_phase_status</code>	35
5.9.1.53	<code>tc_set_start_value</code>	35
5.9.1.54	<code>tc_species_status</code>	35
5.9.1.55	<code>tc_version</code>	35
5.10	<code>tcExamples.c</code> File Reference	36
5.10.1	Typedef Documentation	36

5.10.1.1	ivect	36
5.10.1.2	rvect	36
5.10.1.3	str8	36
5.10.1.4	strvect	36
5.10.2	Function Documentation	36
5.10.2.1	example1	36
5.10.2.2	example2	36
5.10.2.3	example3	37
5.11	tcExamples.h File Reference	37
5.11.1	Function Documentation	37
5.11.1.1	example1	37
5.11.1.2	example2	37
5.11.1.3	example3	37
5.12	tcMain.c File Reference	37
5.12.1	Macro Definition Documentation	38
5.12.1.1	TC_API_LIBRARY_NAME	38
5.12.2	Function Documentation	38
5.12.2.1	importLibThermoCalc	38
5.12.2.2	loadTCLibraryInCurrentDir	38
5.12.2.3	main	38
5.13	tcutils.c File Reference	38
5.13.1	Function Documentation	38
5.13.1.1	getTempEnvironmentPath	38
5.13.1.2	getThermoCalcEnvironmentPath	38
5.14	tcutils.h File Reference	39
5.14.1	Macro Definition Documentation	39
5.14.1.1	getCurrentWorkingDir	39
5.14.1.2	SLASH	39
5.14.1.3	TCDEV_HOME	39
5.14.1.4	TCPATH	39
5.14.1.5	TEMP	39
5.14.2	Function Documentation	39
5.14.2.1	getTempEnvironmentPath	39
5.14.2.2	getThermoCalcEnvironmentPath	40
<b>Index</b>		<b>41</b>

# Chapter 1

## Thermo-Calc c-api

The main part of this manual is a technical description of the TC-API. To find details on the Thermodynamic applications of each library function see the section on [tcapi.h](#).

### 1.1 Installed files

In the distribution of Thermo-Calc c-api, the following folders and files can be found.

#### 1.1.1 TC-API libraries.

These could be .lib, .dll, .so -files depending on your installation.  
E.g. on a 64 bit Windows system you will find:  
tcapi-win-x64-.dll and tcapi-win-x64-.lib

#### 1.1.2 Source folder.

This is the c code for running the the different examples. Some of this code can be reused in user-written projects.

#### 1.1.3 Project folders for building the example code

Linux:

-> Linux/Linux-Dynamic-Linking

-> Linux/Linux-Explicit-Loading

Windows:

-> Windows-Mingw32-Explicit-Loading

-> Windows-Studio-Project-Dynamic-Linking

-> Windows-Studio-Project-Explicit-Loading

## 1.2 Explicit loading or Dynamic linking

When using Dynamic Linking, the libraries (.so or .lib) and the header-files of the tcapi are needed when the program is being built.

When using Explicit Loading no libraries are needed for the build. They are loaded at runtime (.so or .dll).

## 1.3 About the source code

### 1.3.1 The dynamic linking examples use the files

Simple thermodynamic calculations to demonstrate the use of this api.

[example1.c](#)

[example2.c](#)

[example3.c](#)

Utility functions for finding the correct environment variables

[tcutils.c](#)

[tcutils.h](#)

Declarations and documentation on all TC-API functions

[tcapi.h](#)

Thermo-Calc proprietary declarations and definitions. DO NOT EDIT.

[tc\\_data\\_defs.h](#)

### 1.3.2 The explicit loading examples use the files

Utilities to simplify working with explicitly loaded dll.

[libtc.c](#)

[libtc.h](#)

Simple thermodynamic calculations to demonstrate the use of this api.

[tcExamples.c](#)

[tcExamples.h](#)

Main program to exemplify how this api can be used

[tcMain.c](#)

Utility functions for finding the correct environment variables

[tcutils.c](#)

[tcutils.h](#)

Thermo-Calc proprietary declarations and definitions. DO NOT EDIT.

[tc\\_data\\_defs.h](#)

## 1.4 When starting developing proprietary projects the following code is needed

### 1.4.1 The dynamic linking examples use the files

[tcapi.h](#)

[tc\\_data\\_defs.h](#)

### 1.4.2 The explicit loading examples use the files

[libtc.c](#)

[libtc.h](#)

[tc\\_data\\_defs.h](#)

([tcMain.c](#) can give an idea of how the above files can be used)





## Chapter 2

# Class Index

### 2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

<a href="#">_tc_function_library</a>	9
<a href="#">tc_components_strings</a>	15
<a href="#">tc_conditions_as_arrays_of_strings</a>	15
<a href="#">tc_constituents_strings</a>	16
<a href="#">tc_databases_strings</a>	16
<a href="#">tc_elements_strings</a>	17
<a href="#">tc_phases_strings</a>	17
<a href="#">tc_reference_strings</a>	17
<a href="#">tc_species_strings</a>	18



# Chapter 3

## File Index

### 3.1 File List

Here is a list of all files with brief descriptions:

<a href="#">example1.c</a>	19
<a href="#">example2.c</a>	19
<a href="#">example3.c</a>	19
<a href="#">libtc.c</a>	20
<a href="#">libtc.h</a>	21
<a href="#">tc_data_defs.h</a>	24
<a href="#">tcapi.h</a>	29
<a href="#">tcExamples.c</a>	36
<a href="#">tcExamples.h</a>	37
<a href="#">tcMain.c</a>	37
<a href="#">tcutils.c</a>	38
<a href="#">tcutils.h</a>	39



## Chapter 4

# Class Documentation

### 4.1 `_tc_function_library` Struct Reference

```
#include <libtc.h>
```

#### Public Attributes

- `VoidFuncString tc_append_database`
- `BoolFuncStringStringStrLen tc_check_license`
- `StringFuncString tc_component_status`
- `VoidFuncNoParams tc_compute_equilibrium`
- `VoidFuncInt tc_create_new_equilibrium`
- `IntFuncStringInt tc_database`
- `VoidFuncStringIntInt tc_define_components`
- `IntFuncNoParams tc_degrees_of_freedom`
- `VoidFuncNoParams tc_deinit`
- `VoidFuncString tc_delete_condition`
- `VoidFuncString tc_delete_symbol`
- `IntFuncStringInt tc_element`
- `VoidFuncString tc_element_reject`
- `VoidFuncString tc_element_select`
- `VoidFuncStringString tc_enter_ges5_parameter`
- `VoidFuncStringStringIntIntFloatString tc_enter_symbol`
- `BoolFuncIntPStringInt tc_error`
- `VoidFuncString tc_ges5`
- `VoidFuncNoParams tc_get_data`
- `VoidFuncStringFloatPFloatP tc_get_derivatives`
- `VoidFuncStringStringInt tc_get_ges5_parameter`
- `FloatFuncString tc_get_value`
- `IntFuncNoParams tc_init_root`
- `IntFuncStringString tc_init_root3`
- `VoidFuncStringInt tc_list_component`
- `VoidFuncStringInt tc_list_conditions`
- `VoidFuncStringInt tc_list_phase`
- `VoidFuncStringInt tc_list_species`
- `IntFuncStringIntIntP tc_list_symbols`
- `IntFuncString tc_nr_of_constituents_in_phase`
- `VoidFuncString tc_open_database`
- `IntFuncStringInt tc_phase`

- [IntFuncStringIntPStringIntFloatP tc\\_phase\\_all\\_constituents](#)
- [IntFuncStringIntPStringIntFloatP tc\\_phase\\_constituents](#)
- [VoidFuncString tc\\_phase\\_reject](#)
- [VoidFuncString tc\\_phase\\_select](#)
- [StringFuncString tc\\_phase\\_status](#)
- [IntFuncStringIntPStringStrLenFloatP tc\\_phase\\_structure](#)
- [VoidFuncString tc\\_poly3](#)
- [VoidFuncStringFloatP tc\\_put\\_sitefractions](#)
- [VoidFuncString tc\\_read\\_poly3\\_file](#)
- [VoidFuncStringIntString tc\\_reject\\_constituent](#)
- [VoidFuncNoParams tc\\_reset\\_error](#)
- [VoidFuncStringIntString tc\\_restore\\_constituent](#)
- [VoidFuncString tc\\_save\\_poly3\\_file](#)
- [VoidFuncInt tc\\_select\\_equilibrium](#)
- [VoidFuncStringString tc\\_set\\_component\\_status](#)
- [VoidFuncStringFloat tc\\_set\\_condition](#)
- [VoidFuncInt tc\\_set\\_license\\_code](#)
- [VoidFuncIntPIntPIntPIntP tc\\_set\\_minimization\\_option](#)
- [VoidFuncStringFloat tc\\_set\\_phase\\_addition](#)
- [VoidFuncStringStringFloat tc\\_set\\_phase\\_status](#)
- [VoidFuncStringFloat tc\\_set\\_start\\_value](#)
- [StringFuncString tc\\_species\\_status](#)
- [VoidFuncStringInt tc\\_version](#)

#### 4.1.1 Detailed Description

Definition at line 67 of file libtc.h.

#### 4.1.2 Member Data Documentation

##### 4.1.2.1 VoidFuncString \_tc\_function\_library::tc\_append\_database

Definition at line 68 of file libtc.h.

##### 4.1.2.2 BoolFuncStringStringStrLen \_tc\_function\_library::tc\_check\_license

Definition at line 69 of file libtc.h.

##### 4.1.2.3 StringFuncString \_tc\_function\_library::tc\_component\_status

Definition at line 70 of file libtc.h.

##### 4.1.2.4 VoidFuncNoParams \_tc\_function\_library::tc\_compute\_equilibrium

Definition at line 71 of file libtc.h.

##### 4.1.2.5 VoidFuncInt \_tc\_function\_library::tc\_create\_new\_equilibrium

Definition at line 72 of file libtc.h.

#### 4.1.2.6 `IntFuncStringInt _tc_function_library::tc_database`

Definition at line 73 of file `libtc.h`.

#### 4.1.2.7 `VoidFuncStringIntInt _tc_function_library::tc_define_components`

Definition at line 74 of file `libtc.h`.

#### 4.1.2.8 `IntFuncNoParams _tc_function_library::tc_degrees_of_freedom`

Definition at line 75 of file `libtc.h`.

#### 4.1.2.9 `VoidFuncNoParams _tc_function_library::tc_deinit`

Definition at line 76 of file `libtc.h`.

#### 4.1.2.10 `VoidFuncString _tc_function_library::tc_delete_condition`

Definition at line 77 of file `libtc.h`.

#### 4.1.2.11 `VoidFuncString _tc_function_library::tc_delete_symbol`

Definition at line 78 of file `libtc.h`.

#### 4.1.2.12 `IntFuncStringInt _tc_function_library::tc_element`

Definition at line 79 of file `libtc.h`.

#### 4.1.2.13 `VoidFuncString _tc_function_library::tc_element_reject`

Definition at line 80 of file `libtc.h`.

#### 4.1.2.14 `VoidFuncString _tc_function_library::tc_element_select`

Definition at line 81 of file `libtc.h`.

#### 4.1.2.15 `VoidFuncStringString _tc_function_library::tc_enter_ges5_parameter`

Definition at line 82 of file `libtc.h`.

#### 4.1.2.16 `VoidFuncStringStringIntIntFloatString _tc_function_library::tc_enter_symbol`

Definition at line 83 of file `libtc.h`.

#### 4.1.2.17 `BoolFuncIntPStringInt _tc_function_library::tc_error`

Definition at line 84 of file `libtc.h`.



**4.1.2.18 VoidFuncString \_tc\_function\_library::tc\_ges5**

Definition at line 85 of file libtc.h.

**4.1.2.19 VoidFuncNoParams \_tc\_function\_library::tc\_get\_data**

Definition at line 86 of file libtc.h.

**4.1.2.20 VoidFuncStringFloatPFloatP \_tc\_function\_library::tc\_get\_derivatives**

Definition at line 87 of file libtc.h.

**4.1.2.21 VoidFuncStringStringInt \_tc\_function\_library::tc\_get\_ges5\_parameter**

Definition at line 88 of file libtc.h.

**4.1.2.22 FloatFuncString \_tc\_function\_library::tc\_get\_value**

Definition at line 89 of file libtc.h.

**4.1.2.23 IntFuncNoParams \_tc\_function\_library::tc\_init\_root**

Definition at line 90 of file libtc.h.

**4.1.2.24 IntFuncStringString \_tc\_function\_library::tc\_init\_root3**

Definition at line 91 of file libtc.h.

**4.1.2.25 VoidFuncStringInt \_tc\_function\_library::tc\_list\_component**

Definition at line 92 of file libtc.h.

**4.1.2.26 VoidFuncStringInt \_tc\_function\_library::tc\_list\_conditions**

Definition at line 93 of file libtc.h.

**4.1.2.27 VoidFuncStringInt \_tc\_function\_library::tc\_list\_phase**

Definition at line 94 of file libtc.h.

**4.1.2.28 VoidFuncStringInt \_tc\_function\_library::tc\_list\_species**

Definition at line 95 of file libtc.h.

**4.1.2.29 IntFuncStringIntIntP \_tc\_function\_library::tc\_list\_symbols**

Definition at line 96 of file libtc.h.

**4.1.2.30** `IntFuncString _tc_function_library::tc_nr_of_constituents_in_phase`

Definition at line 97 of file `libtc.h`.

**4.1.2.31** `VoidFuncString _tc_function_library::tc_open_database`

Definition at line 98 of file `libtc.h`.

**4.1.2.32** `IntFuncStringInt _tc_function_library::tc_phase`

Definition at line 99 of file `libtc.h`.

**4.1.2.33** `IntFuncStringIntPStringIntFloatP _tc_function_library::tc_phase_all_constituents`

Definition at line 100 of file `libtc.h`.

**4.1.2.34** `IntFuncStringIntPStringIntFloatP _tc_function_library::tc_phase_constituents`

Definition at line 101 of file `libtc.h`.

**4.1.2.35** `VoidFuncString _tc_function_library::tc_phase_reject`

Definition at line 102 of file `libtc.h`.

**4.1.2.36** `VoidFuncString _tc_function_library::tc_phase_select`

Definition at line 103 of file `libtc.h`.

**4.1.2.37** `StringFuncString _tc_function_library::tc_phase_status`

Definition at line 104 of file `libtc.h`.

**4.1.2.38** `IntFuncStringIntPStringStrLenFloatP _tc_function_library::tc_phase_structure`

Definition at line 105 of file `libtc.h`.

**4.1.2.39** `VoidFuncString _tc_function_library::tc_poly3`

Definition at line 106 of file `libtc.h`.

**4.1.2.40** `VoidFuncStringFloatP _tc_function_library::tc_put_sitefractions`

Definition at line 107 of file `libtc.h`.

**4.1.2.41** `VoidFuncString _tc_function_library::tc_read_poly3_file`

Definition at line 108 of file `libtc.h`.

#### 4.1.2.42 VoidFuncStringIntString \_tc\_function\_library::tc\_reject\_constituent

Definition at line 109 of file libtc.h.

#### 4.1.2.43 VoidFuncNoParams \_tc\_function\_library::tc\_reset\_error

Definition at line 110 of file libtc.h.

#### 4.1.2.44 VoidFuncStringIntString \_tc\_function\_library::tc\_restore\_constituent

Definition at line 111 of file libtc.h.

#### 4.1.2.45 VoidFuncString \_tc\_function\_library::tc\_save\_poly3\_file

Definition at line 112 of file libtc.h.

#### 4.1.2.46 VoidFuncInt \_tc\_function\_library::tc\_select\_equilibrium

Definition at line 113 of file libtc.h.

#### 4.1.2.47 VoidFuncStringString \_tc\_function\_library::tc\_set\_component\_status

Definition at line 114 of file libtc.h.

#### 4.1.2.48 VoidFuncStringFloat \_tc\_function\_library::tc\_set\_condition

Definition at line 115 of file libtc.h.

#### 4.1.2.49 VoidFuncInt \_tc\_function\_library::tc\_set\_license\_code

Definition at line 116 of file libtc.h.

#### 4.1.2.50 VoidFuncIntPIntPIntPIntP \_tc\_function\_library::tc\_set\_minimization\_option

Definition at line 117 of file libtc.h.

#### 4.1.2.51 VoidFuncStringFloat \_tc\_function\_library::tc\_set\_phase\_addition

Definition at line 118 of file libtc.h.

#### 4.1.2.52 VoidFuncStringStringFloat \_tc\_function\_library::tc\_set\_phase\_status

Definition at line 119 of file libtc.h.

#### 4.1.2.53 VoidFuncStringFloat \_tc\_function\_library::tc\_set\_start\_value

Definition at line 120 of file libtc.h.

#### 4.1.2.54 StringFuncString\_tc\_function\_library::tc\_species\_status

Definition at line 121 of file libtc.h.

#### 4.1.2.55 VoidFuncStringInt\_tc\_function\_library::tc\_version

Definition at line 122 of file libtc.h.

The documentation for this struct was generated from the following file:

- [libtc.h](#)

## 4.2 tc\_components\_strings Struct Reference

```
#include <tc_data_defs.h>
```

### Public Attributes

- char [component](#) [TC\_STRLEN\_COMPONENTS]

### 4.2.1 Detailed Description

Definition at line 111 of file tc\_data\_defs.h.

### 4.2.2 Member Data Documentation

#### 4.2.2.1 char tc\_components\_strings::component[TC\_STRLEN\_COMPONENTS]

Definition at line 113 of file tc\_data\_defs.h.

The documentation for this struct was generated from the following file:

- [tc\\_data\\_defs.h](#)

## 4.3 tc\_conditions\_as\_arrays\_of\_strings Struct Reference

```
#include <tc_data_defs.h>
```

### Public Attributes

- char [condition](#) [TC\_STRLEN\_MAX]

### 4.3.1 Detailed Description

Definition at line 98 of file tc\_data\_defs.h.

### 4.3.2 Member Data Documentation

#### 4.3.2.1 `char tc_conditions_as_arrays_of_strings::condition[TC_STRLEN_MAX]`

Definition at line 100 of file `tc_data_defs.h`.

The documentation for this struct was generated from the following file:

- [tc\\_data\\_defs.h](#)

## 4.4 `tc_constituents_strings` Struct Reference

```
#include <tc_data_defs.h>
```

### Public Attributes

- `char constituent [TC_STRLEN_CONSTITUENTS]`

#### 4.4.1 Detailed Description

Definition at line 130 of file `tc_data_defs.h`.

#### 4.4.2 Member Data Documentation

##### 4.4.2.1 `char tc_constituents_strings::constituent[TC_STRLEN_CONSTITUENTS]`

Definition at line 132 of file `tc_data_defs.h`.

The documentation for this struct was generated from the following file:

- [tc\\_data\\_defs.h](#)

## 4.5 `tc_databases_strings` Struct Reference

```
#include <tc_data_defs.h>
```

### Public Attributes

- `char database [TC_STRLEN_DATABASE]`

#### 4.5.1 Detailed Description

Definition at line 136 of file `tc_data_defs.h`.

#### 4.5.2 Member Data Documentation

##### 4.5.2.1 `char tc_databases_strings::database[TC_STRLEN_DATABASE]`

Definition at line 138 of file `tc_data_defs.h`.

The documentation for this struct was generated from the following file:

- [tc\\_data\\_defs.h](#)

## 4.6 tc\_elements\_strings Struct Reference

```
#include <tc_data_defs.h>
```

### Public Attributes

- char [element](#) [TC\_STRLEN\_ELEMENTS]

#### 4.6.1 Detailed Description

Definition at line 105 of file tc\_data\_defs.h.

#### 4.6.2 Member Data Documentation

##### 4.6.2.1 char tc\_elements\_strings::element[TC\_STRLEN\_ELEMENTS]

Definition at line 107 of file tc\_data\_defs.h.

The documentation for this struct was generated from the following file:

- [tc\\_data\\_defs.h](#)

## 4.7 tc\_phases\_strings Struct Reference

```
#include <tc_data_defs.h>
```

### Public Attributes

- char [phase](#) [TC\_STRLEN\_PHASES]

#### 4.7.1 Detailed Description

Definition at line 124 of file tc\_data\_defs.h.

#### 4.7.2 Member Data Documentation

##### 4.7.2.1 char tc\_phases\_strings::phase[TC\_STRLEN\_PHASES]

Definition at line 126 of file tc\_data\_defs.h.

The documentation for this struct was generated from the following file:

- [tc\\_data\\_defs.h](#)

## 4.8 tc\_reference\_strings Struct Reference

```
#include <tc_data_defs.h>
```

## Public Attributes

- char [reference](#) [TC\_STRLEN\_REFERENCE]

### 4.8.1 Detailed Description

Definition at line 142 of file tc\_data\_defs.h.

### 4.8.2 Member Data Documentation

#### 4.8.2.1 char tc\_reference\_strings::reference[TC\_STRLEN\_REFERENCE]

Definition at line 144 of file tc\_data\_defs.h.

The documentation for this struct was generated from the following file:

- [tc\\_data\\_defs.h](#)

## 4.9 tc\_species\_strings Struct Reference

```
#include <tc_data_defs.h>
```

## Public Attributes

- char [specie](#) [TC\_STRLEN\_SPECIES]

### 4.9.1 Detailed Description

Definition at line 117 of file tc\_data\_defs.h.

### 4.9.2 Member Data Documentation

#### 4.9.2.1 char tc\_species\_strings::specie[TC\_STRLEN\_SPECIES]

Definition at line 119 of file tc\_data\_defs.h.

The documentation for this struct was generated from the following file:

- [tc\\_data\\_defs.h](#)

## Chapter 5

# File Documentation

### 5.1 example1.c File Reference

```
#include <stdio.h>
#include <string.h>
#include "tcapi.h"
#include "tcutils.h"
Include dependency graph for example1.c:
```

### 5.2 example2.c File Reference

```
#include <stdio.h>
#include <string.h>
#include "tcapi.h"
#include "tcutils.h"
Include dependency graph for example2.c:
```

### Functions

- int [main](#) (int argc, char \*\*argv)

#### 5.2.1 Function Documentation

##### 5.2.1.1 int main ( int argc, char \*\* argv )

Definition at line 14 of file example2.c.

### 5.3 example3.c File Reference

```
#include <stdio.h>
#include <string.h>
#include "tcapi.h"
#include "tcutils.h"
Include dependency graph for example3.c:
```



## Typedefs

- typedef char [str8](#)[8]
- typedef [str8](#) [strvect](#)[100]
- typedef [TC\\_INT](#) [ivect](#)[50]
- typedef [TC\\_FLOAT](#) [rvect](#)[50]

## Functions

- int [main](#) (int argc, char \*\*argv)

### 5.3.1 Typedef Documentation

#### 5.3.1.1 typedef [TC\\_INT](#) [ivect](#)[50]

Definition at line 16 of file [example3.c](#).

#### 5.3.1.2 typedef [TC\\_FLOAT](#) [rvect](#)[50]

Definition at line 17 of file [example3.c](#).

#### 5.3.1.3 typedef char [str8](#)[8]

Definition at line 14 of file [example3.c](#).

#### 5.3.1.4 typedef [str8](#) [strvect](#)[100]

Definition at line 15 of file [example3.c](#).

### 5.3.2 Function Documentation

#### 5.3.2.1 int [main](#) ( int *argc*, char \*\* *argv* )

Definition at line 19 of file [example3.c](#).

## 5.4 libtc.c File Reference

```
#include "libtc.h"  
Include dependency graph for libtc.c:
```

## Functions

- void \* [tloadfunc](#) ([TCHANDLE](#) *tcHandle*, char \**function\_name*)
- int [importFunctions](#) ([TCHANDLE](#) *tcHandle*, [tc\\_function\\_library](#) \**tc*, char \**message*)

### 5.4.1 Function Documentation

#### 5.4.1.1 int [importFunctions](#) ( [TCHANDLE](#) *tcHandle*, [tc\\_function\\_library](#) \* *tc*, char \* *message* )

Definition at line 23 of file [libtc.c](#).

### 5.4.1.2 void\* tloadfunc ( TCHANDLE tcHandle, char \* function\_name )

Definition at line 11 of file libtc.c.

## 5.5 libtc.h File Reference

```
#include "tc_data_defs.h"
#include <dlfcn.h>
#include <errno.h>
#include <string.h>
```

Include dependency graph for libtc.h: This graph shows which files directly or indirectly include this file:

### Classes

- struct [\\_tc\\_function\\_library](#)

### Typedefs

- typedef void(\* [VoidFuncNoParams](#)) ()
- typedef void(\* [VoidFuncString](#)) (TC\_STRING)
- typedef void(\* [VoidFuncInt](#)) (TC\_INT)
- typedef void(\* [VoidFuncStringString](#)) (TC\_STRING, TC\_STRING)
- typedef void(\* [VoidFuncStringFloat](#)) (TC\_STRING, TC\_FLOAT)
- typedef void(\* [VoidFuncStringFloatP](#)) (TC\_STRING, TC\_FLOAT \*)
- typedef void(\* [VoidFuncStringInt](#)) (TC\_STRING, TC\_INT)
- typedef void(\* [VoidFuncStringStringInt](#)) (TC\_STRING, TC\_STRING, TC\_INT)
- typedef void(\* [VoidFuncStringStringFloat](#)) (TC\_STRING, TC\_STRING, TC\_FLOAT)
- typedef void(\* [VoidFuncStringIntInt](#)) (TC\_STRING, TC\_INT, TC\_INT)
- typedef void(\* [VoidFuncStringFloatPFloatP](#)) (TC\_STRING, TC\_FLOAT \*, TC\_FLOAT \*)
- typedef void(\* [VoidFuncStringIntString](#)) (TC\_STRING, TC\_INT, TC\_STRING)
- typedef void(\* [VoidFuncIntPIntPIntP](#)) (TC\_INT \*, TC\_INT \*, TC\_INT \*, TC\_INT \*)
- typedef void(\* [VoidFuncStringStringIntIntFloatString](#)) (TC\_STRING, TC\_STRING, TC\_INT, TC\_INT, TC\_FLOAT, TC\_STRING)
- typedef [TC\\_INT](#)(\* [IntFuncNoParams](#)) ()
- typedef [TC\\_INT](#)(\* [IntFuncString](#)) (TC\_STRING)
- typedef [TC\\_INT](#)(\* [IntFuncStringString](#)) (TC\_STRING, TC\_STRING)
- typedef [TC\\_INT](#)(\* [IntFuncStringInt](#)) (TC\_STRING, TC\_INT)
- typedef [TC\\_INT](#)(\* [IntFuncStringIntIntP](#)) (TC\_STRING, TC\_INT, TC\_INT \*)
- typedef [TC\\_INT](#)(\* [IntFuncStringIntPStringStrLenFloatP](#)) (TC\_STRING, TC\_INT \*, TC\_STRING, TC\_STRING\_LENGTH, TC\_FLOAT \*)
- typedef [TC\\_INT](#)(\* [IntFuncStringIntPStringIntFloatP](#)) (TC\_STRING, TC\_INT \*, TC\_STRING, TC\_INT, TC\_FLOAT \*)
- typedef [TC\\_FLOAT](#)(\* [FloatFuncString](#)) (TC\_STRING)
- typedef [TC\\_STRING](#)(\* [StringFuncString](#)) (TC\_STRING)
- typedef [TC\\_BOOL](#)(\* [BoolFuncIntPStringInt](#)) (TC\_INT \*, TC\_STRING, TC\_INT)
- typedef [TC\\_BOOL](#)(\* [BoolFuncStringStringStrLen](#)) (TC\_STRING, TC\_STRING, TC\_STRING\_LENGTH)
- typedef struct [\\_tc\\_function\\_library](#) [tc\\_function\\_library](#)

### Functions

- int [importFunctions](#) (TCHANDLE tcHandle, [tc\\_function\\_library](#) \*tc, char \*message)

## 5.5.1 Typedef Documentation

### 5.5.1.1 typedef TC\_BOOL(\* BoolFuncIntPStringInt) (TC\_INT \*, TC\_STRING, TC\_INT)

Definition at line 55 of file libtc.h.

### 5.5.1.2 typedef TC\_BOOL(\* BoolFuncStringStringStrLen) (TC\_STRING, TC\_STRING, TC\_STRING\_LENGTH)

Definition at line 56 of file libtc.h.

### 5.5.1.3 typedef TC\_FLOAT(\* FloatFuncString) (TC\_STRING)

Definition at line 51 of file libtc.h.

### 5.5.1.4 typedef TC\_INT(\* IntFuncNoParams) ()

Definition at line 43 of file libtc.h.

### 5.5.1.5 typedef TC\_INT(\* IntFuncString) (TC\_STRING)

Definition at line 44 of file libtc.h.

### 5.5.1.6 typedef TC\_INT(\* IntFuncStringInt) (TC\_STRING, TC\_INT)

Definition at line 46 of file libtc.h.

### 5.5.1.7 typedef TC\_INT(\* IntFuncStringIntIntP) (TC\_STRING, TC\_INT, TC\_INT \*)

Definition at line 47 of file libtc.h.

### 5.5.1.8 typedef TC\_INT(\* IntFuncStringIntPStringIntFloatP) (TC\_STRING, TC\_INT \*, TC\_STRING, TC\_INT, TC\_FLOAT \*)

Definition at line 49 of file libtc.h.

### 5.5.1.9 typedef TC\_INT(\* IntFuncStringIntPStringStrLenFloatP) (TC\_STRING, TC\_INT \*, TC\_STRING, TC\_STRING\_LENGTH, TC\_FLOAT \*)

Definition at line 48 of file libtc.h.

### 5.5.1.10 typedef TC\_INT(\* IntFuncStringString) (TC\_STRING, TC\_STRING)

Definition at line 45 of file libtc.h.

### 5.5.1.11 typedef TC\_STRING(\* StringFuncString) (TC\_STRING)

Definition at line 53 of file libtc.h.

5.5.1.12 `typedef struct _tc_function_library tc_function_library`

5.5.1.13 `typedef void(* VoidFuncInt) (TC_INT)`

Definition at line 27 of file libtc.h.

5.5.1.14 `typedef void(* VoidFuncIntPIntPIntP) (TC_INT *, TC_INT *, TC_INT *, TC_INT *)`

Definition at line 40 of file libtc.h.

5.5.1.15 `typedef void(* VoidFuncNoParams) ()`

Definition at line 24 of file libtc.h.

5.5.1.16 `typedef void(* VoidFuncString) (TC_STRING)`

Definition at line 26 of file libtc.h.

5.5.1.17 `typedef void(* VoidFuncStringFloat) (TC_STRING, TC_FLOAT)`

Definition at line 30 of file libtc.h.

5.5.1.18 `typedef void(* VoidFuncStringFloatP) (TC_STRING, TC_FLOAT *)`

Definition at line 31 of file libtc.h.

5.5.1.19 `typedef void(* VoidFuncStringFloatPFloatP) (TC_STRING, TC_FLOAT *, TC_FLOAT *)`

Definition at line 37 of file libtc.h.

5.5.1.20 `typedef void(* VoidFuncStringInt) (TC_STRING, TC_INT)`

Definition at line 32 of file libtc.h.

5.5.1.21 `typedef void(* VoidFuncStringIntInt) (TC_STRING, TC_INT, TC_INT)`

Definition at line 36 of file libtc.h.

5.5.1.22 `typedef void(* VoidFuncStringIntString) (TC_STRING, TC_INT, TC_STRING)`

Definition at line 38 of file libtc.h.

5.5.1.23 `typedef void(* VoidFuncStringString) (TC_STRING, TC_STRING)`

Definition at line 29 of file libtc.h.

5.5.1.24 `typedef void(* VoidFuncStringStringFloat) (TC_STRING, TC_STRING, TC_FLOAT)`

Definition at line 35 of file libtc.h.

5.5.1.25 `typedef void(* VoidFuncStringStringInt) (TC_STRING, TC_STRING, TC_INT)`

Definition at line 34 of file libtc.h.

5.5.1.26 `typedef void(* VoidFuncStringStringIntIntFloatString) (TC_STRING, TC_STRING, TC_INT, TC_INT, TC_FLOAT, TC_STRING)`

Definition at line 41 of file libtc.h.

## 5.5.2 Function Documentation

5.5.2.1 `int importFunctions ( TCHANDLE tcHandle, tc_function_library * tc, char * message )`

Definition at line 23 of file libtc.c.

## 5.6 ReadMe.txt File Reference

## 5.7 ReadMe.txt File Reference

## 5.8 tc\_data\_defs.h File Reference

This graph shows which files directly or indirectly include this file:

### Classes

- struct [tc\\_conditions\\_as\\_arrays\\_of\\_strings](#)
- struct [tc\\_elements\\_strings](#)
- struct [tc\\_components\\_strings](#)
- struct [tc\\_species\\_strings](#)
- struct [tc\\_phases\\_strings](#)
- struct [tc\\_constituents\\_strings](#)
- struct [tc\\_databases\\_strings](#)
- struct [tc\\_reference\\_strings](#)

### Macros

- `#define TCHANDLE void*`
- `#define TC_NWSG 4000000`
- `#define TC_NWSE 500000`
- `#define TC_STRLEN_SPECIES 25`
- `#define TC_STRLEN_PHASES 25`
- `#define TC_STRLEN_ELEMENTS 3`
- `#define TC_STRLEN_COMPONENTS 25`
- `#define TC_STRLEN_CONSTITUENTS 25`
- `#define TC_STRLEN_DATABASE 9`
- `#define TC_STRLEN_STOICHIOMETRY 81`
- `#define TC_STRLEN_MAX 256`
- `#define TC_STRLEN_PATH_MAX 512`
- `#define TC_STRLEN_REFERENCE 1024`
- `#define TC_MAX_NR_OF_ELEMENTS 40`

- #define TC\_MAX\_NR\_OF\_SPECIES 5000
- #define TC\_MAX\_NR\_OF\_SUBLATTICES 10
- #define TC\_MAX\_NR\_OF\_CONSTITUENTS 200
- #define TC\_MAX\_NR\_OF\_CONST\_PER\_SUBLATTICE 200
- #define TC\_MAX\_NR\_OF\_CONST\_PER\_SUBLATTICE\_IN\_IDEAL\_GAS 5000
- #define TC\_MAX\_NR\_OF\_DATABASES 130
- #define TC\_MAX\_NR\_OF\_AXES 5
- #define TC\_MAX\_NR\_OF\_PHASES 4000 /\* check ITDBPX in tdbmax.inc \*/
- #define TC\_EPS 1.00E-8
- #define TC\_VARS
- #define true 1
- #define false 0
- #define DIExport \_\_declspec( dllexport )
- #define TCFuncExport extern DIExport
- #define INTEGER\_FUNC extern TC\_INT
- #define INTEGER\_FUNC\_WIN INTEGER\_FUNC
- #define INTEGER\_FUNC\_GNU INTEGER\_FUNC
- #define VOID\_FUNC\_WIN extern void
- #define BOOL\_FUNC\_WIN extern TC\_BOOL
- #define FLOAT\_FUNC\_WIN extern TC\_FLOAT

## Typedefs

- typedef long TC\_INT
- typedef long pointer
- typedef double TC\_FLOAT
- typedef TC\_INT TC\_BOOL
- typedef char \* TC\_STRING
- typedef long TC\_STRING\_LENGTH
- typedef struct tc\_conditions\_as\_arrays\_of\_strings tc\_conditions\_as\_arrays\_of\_strings
- typedef struct tc\_elements\_strings tc\_elements\_strings
- typedef struct tc\_components\_strings tc\_components\_strings
- typedef struct tc\_species\_strings tc\_species\_strings
- typedef struct tc\_phases\_strings tc\_phases\_strings
- typedef struct tc\_constituents\_strings tc\_constituents\_strings
- typedef struct tc\_databases\_strings tc\_databases\_strings
- typedef struct tc\_reference\_strings tc\_reference\_strings
- typedef TC\_INT TC\_IARR[4]
- typedef char TC\_LABEL\_STRING[127]

### 5.8.1 Macro Definition Documentation

#### 5.8.1.1 #define BOOL\_FUNC\_WIN extern TC\_BOOL

Definition at line 196 of file tc\_data\_defs.h.

#### 5.8.1.2 #define DIExport \_\_declspec( dllexport )

Definition at line 170 of file tc\_data\_defs.h.

#### 5.8.1.3 #define false 0

Definition at line 161 of file tc\_data\_defs.h.

5.8.1.4 `#define FLOAT_FUNC_WIN extern TC_FLOAT`

Definition at line 197 of file `tc_data_defs.h`.

5.8.1.5 `#define INTEGER_FUNC extern TC_INT`

Definition at line 191 of file `tc_data_defs.h`.

5.8.1.6 `#define INTEGER_FUNC_GNU INTEGER_FUNC`

Definition at line 194 of file `tc_data_defs.h`.

5.8.1.7 `#define INTEGER_FUNC_WIN INTEGER_FUNC`

Definition at line 193 of file `tc_data_defs.h`.

5.8.1.8 `#define TC_EPS 1.00E-8`

Definition at line 77 of file `tc_data_defs.h`.

5.8.1.9 `#define TC_MAX_NR_OF_AXES 5`

Definition at line 74 of file `tc_data_defs.h`.

5.8.1.10 `#define TC_MAX_NR_OF_CONST_PER_SUBLATTICE 200`

Definition at line 71 of file `tc_data_defs.h`.

5.8.1.11 `#define TC_MAX_NR_OF_CONST_PER_SUBLATTICE_IN IDEAL_GAS 5000`

Definition at line 72 of file `tc_data_defs.h`.

5.8.1.12 `#define TC_MAX_NR_OF_CONSTITUENTS 200`

Definition at line 70 of file `tc_data_defs.h`.

5.8.1.13 `#define TC_MAX_NR_OF_DATABASES 130`

Definition at line 73 of file `tc_data_defs.h`.

5.8.1.14 `#define TC_MAX_NR_OF_ELEMENTS 40`

Definition at line 67 of file `tc_data_defs.h`.

5.8.1.15 `#define TC_MAX_NR_OF_PHASES 4000 /* check ITDBPX in tdbmax.inc */`

Definition at line 76 of file `tc_data_defs.h`.

5.8.1.16 `#define TC_MAX_NR_OF_SPECIES 5000`

Definition at line 68 of file tc\_data\_defs.h.

5.8.1.17 `#define TC_MAX_NR_OF_SUBLATTICES 10`

Definition at line 69 of file tc\_data\_defs.h.

5.8.1.18 `#define TC_NWSE 500000`

Definition at line 54 of file tc\_data\_defs.h.

5.8.1.19 `#define TC_NWSG 4000000`

Definition at line 53 of file tc\_data\_defs.h.

5.8.1.20 `#define TC_STRLEN_COMPONENTS 25`

Definition at line 59 of file tc\_data\_defs.h.

5.8.1.21 `#define TC_STRLEN_CONSTITUENTS 25`

Definition at line 60 of file tc\_data\_defs.h.

5.8.1.22 `#define TC_STRLEN_DATABASE 9`

Definition at line 61 of file tc\_data\_defs.h.

5.8.1.23 `#define TC_STRLEN_ELEMENTS 3`

Definition at line 58 of file tc\_data\_defs.h.

5.8.1.24 `#define TC_STRLEN_MAX 256`

Definition at line 63 of file tc\_data\_defs.h.

5.8.1.25 `#define TC_STRLEN_PATH_MAX 512`

Definition at line 64 of file tc\_data\_defs.h.

5.8.1.26 `#define TC_STRLEN_PHASES 25`

Definition at line 57 of file tc\_data\_defs.h.

5.8.1.27 `#define TC_STRLEN_REFERENCE 1024`

Definition at line 65 of file tc\_data\_defs.h.



**5.8.1.28 #define TC\_STRLEN\_SPECIES 25**

Definition at line 56 of file tc\_data\_defs.h.

**5.8.1.29 #define TC\_STRLEN\_STOICHIOMETRY 81**

Definition at line 62 of file tc\_data\_defs.h.

**5.8.1.30 #define TC\_VARS**

Definition at line 151 of file tc\_data\_defs.h.

**5.8.1.31 #define TCFuncExport extern DIExport**

Definition at line 182 of file tc\_data\_defs.h.

**5.8.1.32 #define TCHANDLE void\***

Definition at line 36 of file tc\_data\_defs.h.

**5.8.1.33 #define true 1**

Definition at line 157 of file tc\_data\_defs.h.

**5.8.1.34 #define VOID\_FUNC\_WIN extern void**

Definition at line 195 of file tc\_data\_defs.h.

**5.8.2 Typedef Documentation****5.8.2.1 typedef long pointer**

Definition at line 20 of file tc\_data\_defs.h.

**5.8.2.2 typedef TC\_INT TC\_BOOL**

Definition at line 41 of file tc\_data\_defs.h.

**5.8.2.3 typedef struct tc\_components\_strings tc\_components\_strings**

Definition at line 110 of file tc\_data\_defs.h.

**5.8.2.4 typedef struct tc\_conditions\_as\_arrays\_of\_strings tc\_conditions\_as\_arrays\_of\_strings**

Definition at line 97 of file tc\_data\_defs.h.

**5.8.2.5 typedef struct tc\_constituents\_strings tc\_constituents\_strings**

Definition at line 129 of file tc\_data\_defs.h.

#### 5.8.2.6 typedef struct tc\_databases\_strings tc\_databases\_strings

Definition at line 135 of file tc\_data\_defs.h.

#### 5.8.2.7 typedef struct tc\_elements\_strings tc\_elements\_strings

Definition at line 104 of file tc\_data\_defs.h.

#### 5.8.2.8 typedef double TC\_FLOAT

Definition at line 31 of file tc\_data\_defs.h.

#### 5.8.2.9 typedef TC\_INT TC\_IARR[4]

Definition at line 147 of file tc\_data\_defs.h.

#### 5.8.2.10 typedef long TC\_INT

Definition at line 19 of file tc\_data\_defs.h.

#### 5.8.2.11 typedef char TC\_LABEL\_STRING[127]

Definition at line 148 of file tc\_data\_defs.h.

#### 5.8.2.12 typedef struct tc\_phases\_strings tc\_phases\_strings

Definition at line 123 of file tc\_data\_defs.h.

#### 5.8.2.13 typedef struct tc\_reference\_strings tc\_reference\_strings

Definition at line 141 of file tc\_data\_defs.h.

#### 5.8.2.14 typedef struct tc\_species\_strings tc\_species\_strings

Definition at line 116 of file tc\_data\_defs.h.

#### 5.8.2.15 typedef char\* TC\_STRING

Definition at line 42 of file tc\_data\_defs.h.

#### 5.8.2.16 typedef long TC\_STRING\_LENGTH

Definition at line 50 of file tc\_data\_defs.h.

## 5.9 tcapi.h File Reference

```
#include "tc_data_defs.h"
```

Include dependency graph for tcapi.h: This graph shows which files directly or indirectly include this file:

## Functions

- void `tc_append_database` (`TC_STRING` name)
- `TC_BOOL` `tc_check_license` (`TC_STRING` name, `TC_STRING` message, `TC_STRING_LENGTH` strlen, `TC_STRING` message)
- `TC_STRING` `tc_component_status` (`TC_STRING` component\_name)
- void `tc_compute_equilibrium` ()
- void `tc_create_new_equilibrium` (`TC_INT` equilibrium)
- `TC_INT` `tc_database` (`TC_STRING` datan, `TC_INT` linelen)
- void `tc_define_components` (`TC_STRING` component\_name, `TC_INT` strlen, `TC_INT` number\_of\_components)
- `TC_INT` `tc_degrees_of_freedom` ()
- void `tc_deinit` ()
- void `tc_delete_condition` (`TC_STRING` condition)
- void `tc_delete_symbol` (`TC_STRING` symbol)
- `TC_INT` `tc_element` (`TC_STRING` elements, `TC_INT` linelen)
- void `tc_element_reject` (`TC_STRING` element\_name)
- void `tc_element_select` (`TC_STRING` element\_name)
- void `tc_enter_ges5_parameter` (`TC_STRING` parameter, `TC_STRING` expression)
- void `tc_enter_symbol` (`TC_STRING` symbol, `TC_STRING` type, `TC_INT` argument\_type, `TC_INT` integer\_argument, `TC_FLOAT` double\_argument, `TC_STRING` string\_argument)
- `TC_BOOL` `tc_error` (`TC_INT` \*error\_number, `TC_STRING` message, `TC_INT` strlen)
- void `tc_ges5` (`TC_STRING` command)
- void `tc_get_data` ()
- void `tc_get_derivatives` (`TC_STRING` phase\_name, `TC_FLOAT` \*arr1, `TC_FLOAT` \*arr2)
- void `tc_get_ges5_parameter` (`TC_STRING` parameter, `TC_STRING` expression, `TC_INT` strlenExpression)
- `TC_FLOAT` `tc_get_value` (`TC_STRING` symbol)
- `TC_INT` `tc_init_root` ()
- `TC_INT` `tc_init_root3` (`TC_STRING` tmppath, `TC_STRING` tpath)
- `TC_INT` `tc_list_component` (`TC_STRING` component\_name, `TC_INT` strlen)
- `TC_INT` `tc_list_conditions` (`TC_STRING` conditions, `TC_INT` strlen)
- `TC_INT` `tc_list_phase` (`TC_STRING` phase\_name, `TC_INT` strlen)
- `TC_INT` `tc_list_species` (`TC_STRING` species\_name, `TC_INT` strlen)
- `TC_INT` `tc_list_symbols` (`TC_STRING` symbols, `TC_INT` strlen, `TC_INT` \*type)
- `TC_INT` `tc_nr_of_constituents_in_phase` (`TC_STRING` phase\_name)
- void `tc_open_database` (`TC_STRING` name)
- `TC_INT` `tc_phase` (`TC_STRING` phases, `TC_INT` linelen)
- `TC_INT` `tc_phase_all_constituents` (`TC_STRING` phase\_name, `TC_INT` \*constituent\_array, `TC_STRING` element\_array, `TC_INT` strLenElem, `TC_FLOAT` \*number\_of\_sites)
- `TC_INT` `tc_phase_constituents` (`TC_STRING` phase\_name, `TC_INT` \*constituent\_array, `TC_STRING` element\_array, `TC_INT` strLenElem, `TC_FLOAT` \*number\_of\_sites)
- void `tc_phase_reject` (`TC_STRING` phase\_name)
- void `tc_phase_select` (`TC_STRING` phase\_name)
- `TC_STRING` `tc_phase_status` (`TC_STRING` phase\_name)
- `TC_INT` `tc_phase_structure` (`TC_STRING` phase\_name, `TC_INT` \*constituent\_array, `TC_STRING` species\_array, `TC_STRING_LENGTH` strLenSpecies, `TC_FLOAT` \*number\_of\_sites)
- void `tc_poly3` (`TC_STRING` command)
- void `tc_put_sitefractions` (`TC_STRING` phase\_name, `TC_FLOAT` \*sfarr)
- void `tc_read_poly3_file` (`TC_STRING` filename)
- void `tc_reject_constituent` (`TC_STRING` phase\_name, `TC_INT` sublattice, `TC_STRING` constituent)
- void `tc_reset_error` ()
- void `tc_restore_constituent` (`TC_STRING` phase\_name, `TC_INT` sublattice, `TC_STRING` constituent)
- void `tc_save_poly3_file` (`TC_STRING` filename)
- void `tc_select_equilibrium` (`TC_INT` equilibrium)
- void `tc_set_component_status` (`TC_STRING` component\_name, `TC_STRING` status)

- void `tc_set_condition` (`TC_STRING` condition, `TC_FLOAT` value)
- void `tc_set_license_code` (`TC_INT` code)
- void `tc_set_minimization_option` (`TC_INT` \*global\_flag, `TC_INT` \*max\_gridpoints, `TC_INT` \*frequency, `TC_INT` \*mesh\_flag)
- void `tc_set_phase_addition` (`TC_STRING` phase\_name, `TC_FLOAT` addition)
- void `tc_set_phase_status` (`TC_STRING` phase\_name, `TC_STRING` status, `TC_FLOAT` value)
- void `tc_set_start_value` (`TC_STRING` state\_variable, `TC_FLOAT` starting\_value)
- `TC_STRING` `tc_species_status` (`TC_STRING` species\_name)
- void `tc_version` (`TC_STRING` str, `TC_INT` str\_len)

## 5.9.1 Function Documentation

### 5.9.1.1 void `tc_append_database` ( `TC_STRING` name )

Appends the named database

### 5.9.1.2 `TC_BOOL` `tc_check_license` ( `TC_STRING` name, `TC_STRING` message, `TC_STRING_LENGTH` strlen\_message )

Returns true if the checked license is available and valid, currently accepts: `TC_DLL`, `TC_GUI` and `TC_TC4U`

### 5.9.1.3 `TC_STRING` `tc_component_status` ( `TC_STRING` component\_name )

Returns the status of "component\_name" where status may be one of "ENTERED" or "SUSPENDED"

### 5.9.1.4 void `tc_compute_equilibrium` ( )

Computes the equilibrium in POLY-3 using the currently set conditions

### 5.9.1.5 void `tc_create_new_equilibrium` ( `TC_INT` equilibrium )

Creates a new equilibrium in POLY-3 with number "equilibrium number".

### 5.9.1.6 `TC_INT` `tc_database` ( `TC_STRING` datan, `TC_INT` linelen )

Returns the number of databases in the system and their names in "datan"

### 5.9.1.7 void `tc_define_components` ( `TC_STRING` component\_name, `TC_INT` strlen, `TC_INT` number\_of\_components )

Redefines the components in the system to the components in "component\_name".

### 5.9.1.8 `TC_INT` `tc_degrees_of_freedom` ( )

Returns the degrees of freedom in the system. This must be zero in order to perform an equilibrium calculation.

### 5.9.1.9 void `tc_deinit` ( )

### 5.9.1.10 void `tc_delete_condition` ( `TC_STRING` condition )

Deletes the condition for the expression in "condition".

5.9.1.11 void `tc_delete_symbol` ( *TC\_STRING symbol* )

Deletes a symbol in the system.

5.9.1.12 *TC\_INT* `tc_element` ( *TC\_STRING elements*, *TC\_INT lincelen* )

Returns the number of elements in the database and their names in "elements"

5.9.1.13 void `tc_element_reject` ( *TC\_STRING element\_name* )

Rejects "element\_name" in the currently selected database.

5.9.1.14 void `tc_element_select` ( *TC\_STRING element\_name* )

Selects "element\_name" in the currently selected database.

5.9.1.15 void `tc_enter_ges5_parameter` ( *TC\_STRING parameter*, *TC\_STRING expression* )

Enters a parameter expression

5.9.1.16 void `tc_enter_symbol` ( *TC\_STRING symbol*, *TC\_STRING type*, *TC\_INT argument\_type*, *TC\_INT integer\_argument*, *TC\_FLOAT double\_argument*, *TC\_STRING string\_argument* )

Enters a symbol in the system, the symbol type may be one of "CONSTANT", "VARIABLE", "FUNCTION" or "TABLE", "argument\_type" defines which of the following arguments will be used, 1 indicates the integer argument, 2 the double argument and 3 the string argument.

5.9.1.17 *TC\_BOOL* `tc_error` ( *TC\_INT \* error\_number*, *TC\_STRING message*, *TC\_INT strlen* )

Returns true if an error has been set, returning the error number in "error\_number" and its corresponding message in "message"

5.9.1.18 void `tc_ges5` ( *TC\_STRING command* )

Sends a command to the GES5 module as defined in the argument "command"

5.9.1.19 void `tc_get_data` ( )

Executes the database command "GET\_DATA"

5.9.1.20 void `tc_get_derivatives` ( *TC\_STRING phase\_name*, *TC\_FLOAT \* arr1*, *TC\_FLOAT \* arr2* )

Returns Gm and the first derivatives with respect to site-fractions in "arr1" and the second derivatives in "arr2" as GM.Y1.Y1, GM.Y1.Y2, GM.Y2.Y2, GM.Y1.Y3, GM.Y2.Y3, GM.YN.YN

5.9.1.21 void `tc_get_ges5_parameter` ( *TC\_STRING parameter*, *TC\_STRING expression*, *TC\_INT strlenExpression* )

Retrieves the expression of a parameter name

**5.9.1.22 TC\_FLOAT tc\_get\_value ( TC\_STRING *symbol* )**

Retrieves the symbol or state variable value from the POLY-3 module.

**5.9.1.23 TC\_INT tc\_init\_root ( )**

Initializes the Thermo-Calc system. This function (or tc\_init\_root3) must be called prior to anything else.

**5.9.1.24 TC\_INT tc\_init\_root3 ( TC\_STRING *tmppath*, TC\_STRING *tcpath* )**

Initializes the Thermo-Calc system. This function (or tc\_init\_root) must be called prior to anything else. *tmppath* Path to directory for log file *tcpath* Path to Thermo-Calc installation (Used to find databases)

**5.9.1.25 TC\_INT tc\_list\_component ( TC\_STRING *component\_name*, TC\_INT *strlen* )**

Returns the number of components in the system and their names in "*component\_name*".

**5.9.1.26 TC\_INT tc\_list\_conditions ( TC\_STRING *conditions*, TC\_INT *strlen* )**

Returns the number of conditions set and their values in "*conditions*".

**5.9.1.27 TC\_INT tc\_list\_phase ( TC\_STRING *phase\_name*, TC\_INT *strlen* )**

Returns the number of phases in the system and their names in "*phase\_name*".

**5.9.1.28 TC\_INT tc\_list\_species ( TC\_STRING *species\_name*, TC\_INT *strlen* )**

Returns the number of species in the system and their names in "*species\_name*".

**5.9.1.29 TC\_INT tc\_list\_symbols ( TC\_STRING *symbols*, TC\_INT *strlen*, TC\_INT \* *type* )**

Returns the number of defined symbols in the system with their expression and value in "*symbols*" and their corresponding type in "*type of symbol*", where the type may be one of 1="CONSTANT", 2="VARIABLE" 3="FUNCTION" 4="TABLE"

**5.9.1.30 TC\_INT tc\_nr\_of\_constituents\_in\_phase ( TC\_STRING *phase\_name* )****5.9.1.31 void tc\_open\_database ( TC\_STRING *name* )**

Opens the named database "*name\_of\_database*".

**5.9.1.32 TC\_INT tc\_phase ( TC\_STRING *phases*, TC\_INT *linelen* )**

Returns the number of phases in the system with the selected elements. NOTE: the routine returns the number of all available phases.

**5.9.1.33** `TC_INT tc_phase_all_constituents ( TC_STRING phase_name, TC_INT * constituent_array, TC_STRING element_array, TC_INT strLenElem, TC_FLOAT * number_of_sites )`

Returns the number of sublattices in the phase (including phases with the status SUSPENDED), the number of constituents on each sublattice in "constituent\_array", the name of the selected species on each sublattice one after each other in "element\_array" and the "number\_of\_sites" on each sublattice.

**5.9.1.34** `TC_INT tc_phase_constituents ( TC_STRING phase_name, TC_INT * constituent_array, TC_STRING element_array, TC_INT strLenElem, TC_FLOAT * number_of_sites )`

Returns the number of sublattices in the phase, the number of constituents on each sublattice in "constituent\_array", the name of the selected species on each sublattice one after each other in "element\_array" and the "number of sites" on each sublattice.

**5.9.1.35** `void tc_phase_reject ( TC_STRING phase_name )`

Rejects the phase in "phase\_name"

**5.9.1.36** `void tc_phase_select ( TC_STRING phase_name )`

Selects the phase in "phase\_name"

**5.9.1.37** `TC_STRING tc_phase_status ( TC_STRING phase_name )`

Returns the status of "phase\_name" where status may be one of "FIXED", "SUSPENDED" or "ENTERED"

**5.9.1.38** `TC_INT tc_phase_structure ( TC_STRING phase_name, TC_INT * constituent_array, TC_STRING species_array, TC_STRING_LENGTH strLenSpecies, TC_FLOAT * number_of_sites )`

Returns the number of sublattices in the phase, the number of constituents on each sublattice in "constituent\_array", the name of the species on each sublattice one after each other in "species\_array" and the number of sites in "number of sites".

**5.9.1.39** `void tc_poly3 ( TC_STRING command )`

Sends a command to POLY-3 module as defined in the argument "command"

**5.9.1.40** `void tc_put_sitefractions ( TC_STRING phase_name, TC_FLOAT * sfarr )`

**5.9.1.41** `void tc_read_poly3_file ( TC_STRING filename )`

Loads the workspace from file "filename" in POLY-3.

**5.9.1.42** `void tc_reject_constituent ( TC_STRING phase_name, TC_INT sublattice, TC_STRING constituent )`

Rejects the constituent "constituent" on sublattice "sublattice" from phase "phase\_name".

**5.9.1.43** `void tc_reset_error ( )`

Resets the error if an error has been set

5.9.1.44 void `tc_restore_constituent` ( *TC\_STRING phase\_name*, *TC\_INT sublattice*, *TC\_STRING constituent* )

Restores the constituent "constituent" on sublattice "sublattice" from phase "phase\_name".

5.9.1.45 void `tc_save_poly3_file` ( *TC\_STRING filename* )

Stores/overwrites the current workspace in POLY-3 on the file "filename".

5.9.1.46 void `tc_select_equilibrium` ( *TC\_INT equilibrium* )

Selects an equilibrium in POLY-3 with number "equilibrium".

5.9.1.47 void `tc_set_component_status` ( *TC\_STRING component\_name*, *TC\_STRING status* )

Sets the status of "component\_name" to "status" to one of "ENTERED" or "SUSPENDED"

5.9.1.48 void `tc_set_condition` ( *TC\_STRING condition*, *TC\_FLOAT value* )

Sets a condition for the expression in "condition" to value in "value".

5.9.1.49 void `tc_set_license_code` ( *TC\_INT code* )

5.9.1.50 void `tc_set_minimization_option` ( *TC\_INT \* global\_flag*, *TC\_INT \* max\_gridpoints*, *TC\_INT \* frequency*,  
*TC\_INT \* mesh\_flag* )

Sets parameters for global minimization

5.9.1.51 void `tc_set_phase_addition` ( *TC\_STRING phase\_name*, *TC\_FLOAT addition* )

Sets the addition "addition" to the Gibbs

5.9.1.52 void `tc_set_phase_status` ( *TC\_STRING phase\_name*, *TC\_STRING status*, *TC\_FLOAT value* )

Sets the status of "phase\_name" to "status" to one of "FIXED", "SUSPENDED", "DORMANT" or "ENTERED".

5.9.1.53 void `tc_set_start_value` ( *TC\_STRING state\_variable*, *TC\_FLOAT starting\_value* )

Sets a starting value for the "state\_variable" to "start\_value".

5.9.1.54 *TC\_STRING* `tc_species_status` ( *TC\_STRING species\_name* )

Returns the status of "species\_name" where status may be one of "ENTERED" or "SUSPENDED"

5.9.1.55 void `tc_version` ( *TC\_STRING str*, *TC\_INT str\_len* )

Returns the version of Thermo-Calc in "str"



## 5.10 tcExamples.c File Reference

```
#include <stdio.h>
#include "tcExamples.h"
#include "tcutils.h"
Include dependency graph for tcExamples.c:
```

### Typedefs

- typedef char [str8](#)[8]
- typedef [str8](#) [strvect](#)[100]
- typedef [TC\\_INT](#) [ivect](#)[50]
- typedef [TC\\_FLOAT](#) [rvect](#)[50]

### Functions

- void [example1](#) ([tc\\_function\\_library](#) \*[tc](#))
- void [example2](#) ([tc\\_function\\_library](#) \*[tc](#))
- void [example3](#) ([tc\\_function\\_library](#) \*[tc](#))

### 5.10.1 Typedef Documentation

#### 5.10.1.1 typedef [TC\\_INT](#) [ivect](#)[50]

Definition at line 288 of file [tcExamples.c](#).

#### 5.10.1.2 typedef [TC\\_FLOAT](#) [rvect](#)[50]

Definition at line 289 of file [tcExamples.c](#).

#### 5.10.1.3 typedef char [str8](#)[8]

Definition at line 286 of file [tcExamples.c](#).

#### 5.10.1.4 typedef [str8](#) [strvect](#)[100]

Definition at line 287 of file [tcExamples.c](#).

### 5.10.2 Function Documentation

#### 5.10.2.1 void [example1](#) ( [tc\\_function\\_library](#) \* [tc](#) )

[tcExamples.h](#)

Definition at line 9 of file [tcExamples.c](#).

#### 5.10.2.2 void [example2](#) ( [tc\\_function\\_library](#) \* [tc](#) )

Definition at line 201 of file [tcExamples.c](#).

## 5.10.2.3 void example3 ( tc\_function\_library \* tc )

Definition at line 291 of file tcExamples.c.

## 5.11 tcExamples.h File Reference

```
#include "libtc.h"
```

Include dependency graph for tcExamples.h: This graph shows which files directly or indirectly include this file:

### Functions

- void [example1](#) (tc\_function\_library \*tc)
- void [example2](#) (tc\_function\_library \*tc)
- void [example3](#) (tc\_function\_library \*tc)

#### 5.11.1 Function Documentation

## 5.11.1.1 void example1 ( tc\_function\_library \* tc )

[tcExamples.h](#)

Definition at line 9 of file tcExamples.c.

## 5.11.1.2 void example2 ( tc\_function\_library \* tc )

Definition at line 201 of file tcExamples.c.

## 5.11.1.3 void example3 ( tc\_function\_library \* tc )

Definition at line 291 of file tcExamples.c.

## 5.12 tcMain.c File Reference

```
#include <stdio.h>
#include <string.h>
#include "tcExamples.h"
#include "tcutils.h"
```

Include dependency graph for tcMain.c:

### Macros

- #define [TC\\_API\\_LIBRARY\\_NAME](#) "libtcapi-linux-ia32-gfortran-7.8.11410.so"

### Functions

- [TCHANDLE loadTCLibraryInCurrentDir](#) ()
- int [importLibThermoCalc](#) (tc\_function\_library \*tc, char \*message)
- int [main](#) (int argc, char \*argv[ ])

## 5.12.1 Macro Definition Documentation

### 5.12.1.1 `#define TC_API_LIBRARY_NAME "libtcapi-linux-ia32-gfortran-7.8.11410.so"`

Definition at line 28 of file tcMain.c.

## 5.12.2 Function Documentation

### 5.12.2.1 `int importLibThermoCalc ( tc_function_library * tc, char * message )`

Definition at line 66 of file tcMain.c.

### 5.12.2.2 `TCHANDLE loadTCLibraryInCurrentDir ( )`

Definition at line 38 of file tcMain.c.

### 5.12.2.3 `int main ( int argc, char * argv[] )`

Definition at line 100 of file tcMain.c.

## 5.13 tcutils.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "tcutils.h"
Include dependency graph for tcutils.c:
```

### Functions

- void [getThermoCalcEnvironmentPath](#) (char \*pathBuffer)
- void [getTempEnvironmentPath](#) (char \*pathBuffer)

## 5.13.1 Function Documentation

### 5.13.1.1 `void getTempEnvironmentPath ( char * pathBuffer )`

Get path to temp directory If it can't find it - default to current working directory

Definition at line 29 of file tcutils.c.

### 5.13.1.2 `void getThermoCalcEnvironmentPath ( char * pathBuffer )`

Get path from the values of Thermo-Calc environment variables TCDEV\_HOME TCPATH Older versions and fall-back

Definition at line 12 of file tcutils.c.

## 5.14 tcutils.h File Reference

```
#include <unistd.h>
```

Include dependency graph for tcutils.h: This graph shows which files directly or indirectly include this file:

### Macros

- #define [getCurrentWorkingDir](#) getcwd
- #define [TCPATH](#) "TCPATH"
- #define [TCDEV\\_HOME](#) "TCDEV\_HOME"
- #define [TEMP](#) "TMPDIR"
- #define [SLASH](#) "/"

### Functions

- void [getThermoCalcEnvironmentPath](#) (char \*pathBuffer)
- void [getTempEnvironmentPath](#) (char \*pathBuffer)

#### 5.14.1 Macro Definition Documentation

##### 5.14.1.1 #define [getCurrentWorkingDir](#) getcwd

Definition at line 6 of file tcutils.h.

##### 5.14.1.2 #define [SLASH](#) "/"

Definition at line 21 of file tcutils.h.

##### 5.14.1.3 #define [TCDEV\\_HOME](#) "TCDEV\_HOME"

Definition at line 10 of file tcutils.h.

##### 5.14.1.4 #define [TCPATH](#) "TCPATH"

Definition at line 9 of file tcutils.h.

##### 5.14.1.5 #define [TEMP](#) "TMPDIR"

Definition at line 15 of file tcutils.h.

#### 5.14.2 Function Documentation

##### 5.14.2.1 void [getTempEnvironmentPath](#) ( char \* *pathBuffer* )

Get path to temp directory If it can't find it - default to current working directory

Definition at line 29 of file tcutils.c.

#### 5.14.2.2 void `getThermoCalcEnvironmentPath ( char * pathBuffer )`

Get path from the values of Thermo-Calc environment variables TCDEV\_HOME TCPATH Older versions and fall-back

Definition at line 12 of file `tcutils.c`.

# Index

- [\\_tc\\_function\\_library](#), 9
    - [tc\\_append\\_database](#), 10
    - [tc\\_check\\_license](#), 10
    - [tc\\_component\\_status](#), 10
    - [tc\\_compute\\_equilibrium](#), 10
    - [tc\\_create\\_new\\_equilibrium](#), 10
    - [tc\\_database](#), 10
    - [tc\\_define\\_components](#), 11
    - [tc\\_degrees\\_of\\_freedom](#), 11
    - [tc\\_deinit](#), 11
    - [tc\\_delete\\_condition](#), 11
    - [tc\\_delete\\_symbol](#), 11
    - [tc\\_element](#), 11
    - [tc\\_element\\_reject](#), 11
    - [tc\\_element\\_select](#), 11
    - [tc\\_enter\\_ges5\\_parameter](#), 11
    - [tc\\_enter\\_symbol](#), 11
    - [tc\\_error](#), 11
    - [tc\\_ges5](#), 11
    - [tc\\_get\\_data](#), 12
    - [tc\\_get\\_derivatives](#), 12
    - [tc\\_get\\_ges5\\_parameter](#), 12
    - [tc\\_get\\_value](#), 12
    - [tc\\_init\\_root](#), 12
    - [tc\\_init\\_root3](#), 12
    - [tc\\_list\\_component](#), 12
    - [tc\\_list\\_conditions](#), 12
    - [tc\\_list\\_phase](#), 12
    - [tc\\_list\\_species](#), 12
    - [tc\\_list\\_symbols](#), 12
    - [tc\\_nr\\_of\\_constituents\\_in\\_phase](#), 12
    - [tc\\_open\\_database](#), 13
    - [tc\\_phase](#), 13
    - [tc\\_phase\\_all\\_constituents](#), 13
    - [tc\\_phase\\_constituents](#), 13
    - [tc\\_phase\\_reject](#), 13
    - [tc\\_phase\\_select](#), 13
    - [tc\\_phase\\_status](#), 13
    - [tc\\_phase\\_structure](#), 13
    - [tc\\_poly3](#), 13
    - [tc\\_put\\_sitefractions](#), 13
    - [tc\\_read\\_poly3\\_file](#), 13
    - [tc\\_reject\\_constituent](#), 13
    - [tc\\_reset\\_error](#), 14
    - [tc\\_restore\\_constituent](#), 14
    - [tc\\_save\\_poly3\\_file](#), 14
    - [tc\\_select\\_equilibrium](#), 14
    - [tc\\_set\\_component\\_status](#), 14
    - [tc\\_set\\_condition](#), 14
    - [tc\\_set\\_license\\_code](#), 14
    - [tc\\_set\\_minimization\\_option](#), 14
    - [tc\\_set\\_phase\\_addition](#), 14
    - [tc\\_set\\_phase\\_status](#), 14
    - [tc\\_set\\_start\\_value](#), 14
    - [tc\\_species\\_status](#), 14
    - [tc\\_version](#), 15
  - [BOOL\\_FUNC\\_WIN](#)
    - [tc\\_data\\_defs.h](#), 25
  - [BoolFuncIntPStringInt](#)
    - [libtc.h](#), 22
  - [BoolFuncStringStringStrLen](#)
    - [libtc.h](#), 22
  - [component](#)
    - [tc\\_components\\_strings](#), 15
  - [condition](#)
    - [tc\\_conditions\\_as\\_arrays\\_of\\_strings](#), 16
  - [constituent](#)
    - [tc\\_constituents\\_strings](#), 16
  - [database](#)
    - [tc\\_databases\\_strings](#), 16
  - [DllExport](#)
    - [tc\\_data\\_defs.h](#), 25
  - [element](#)
    - [tc\\_elements\\_strings](#), 17
  - [example1](#)
    - [tcExamples.c](#), 36
    - [tcExamples.h](#), 37
  - [example1.c](#), 19
  - [example2](#)
    - [tcExamples.c](#), 36
    - [tcExamples.h](#), 37
  - [example2.c](#), 19
  - [main](#), 19
  - [example3](#)
    - [tcExamples.c](#), 36
    - [tcExamples.h](#), 37
  - [example3.c](#), 19
  - [ivect](#), 20
  - [main](#), 20
  - [rvect](#), 20
  - [str8](#), 20
  - [strvect](#), 20
- [FLOAT\\_FUNC\\_WIN](#)
  - [tc\\_data\\_defs.h](#), 25
- [false](#)

- tc\_data\_defs.h, 25
- FloatFuncString
  - libtc.h, 22
- getCurrentWorkingDir
  - tcutils.h, 39
- getTempEnvironmentPath
  - tcutils.c, 38
  - tcutils.h, 39
- getThermoCalcEnvironmentPath
  - tcutils.c, 38
  - tcutils.h, 39
- INTEGER\_FUNC
  - tc\_data\_defs.h, 26
- INTEGER\_FUNC\_GNU
  - tc\_data\_defs.h, 26
- INTEGER\_FUNC\_WIN
  - tc\_data\_defs.h, 26
- importFunctions
  - libtc.c, 20
  - libtc.h, 24
- importLibThermoCalc
  - tcMain.c, 38
- IntFuncNoParams
  - libtc.h, 22
- IntFuncString
  - libtc.h, 22
- IntFuncStringInt
  - libtc.h, 22
- IntFuncStringIntIntP
  - libtc.h, 22
- IntFuncStringIntPStringIntFloatP
  - libtc.h, 22
- IntFuncStringIntPStringStrLenFloatP
  - libtc.h, 22
- IntFuncStringString
  - libtc.h, 22
- ivect
  - example3.c, 20
  - tcExamples.c, 36
- libtc.c, 20
  - importFunctions, 20
  - tloadfunc, 20
- libtc.h, 21
  - BoolFuncIntPStringInt, 22
  - BoolFuncStringStringStrLen, 22
  - FloatFuncString, 22
  - importFunctions, 24
  - IntFuncNoParams, 22
  - IntFuncString, 22
  - IntFuncStringInt, 22
  - IntFuncStringIntIntP, 22
  - IntFuncStringIntPStringIntFloatP, 22
  - IntFuncStringIntPStringStrLenFloatP, 22
  - IntFuncStringString, 22
  - StringFuncString, 22
  - tc\_function\_library, 22
  - VoidFuncInt, 23
  - VoidFuncIntPIntPIntPIntP, 23
  - VoidFuncNoParams, 23
  - VoidFuncString, 23
  - VoidFuncStringFloat, 23
  - VoidFuncStringFloatP, 23
  - VoidFuncStringFloatPFloatP, 23
  - VoidFuncStringInt, 23
  - VoidFuncStringIntInt, 23
  - VoidFuncStringIntString, 23
  - VoidFuncStringString, 23
  - VoidFuncStringStringFloat, 23
  - VoidFuncStringStringInt, 23
  - VoidFuncStringStringIntIntFloatString, 24
- loadTCLibraryInCurrentDir
  - tcMain.c, 38
- main
  - example2.c, 19
  - example3.c, 20
  - tcMain.c, 38
- phase
  - tc\_phases\_strings, 17
- pointer
  - tc\_data\_defs.h, 28
- ReadMe.txt, 24
- reference
  - tc\_reference\_strings, 18
- rvect
  - example3.c, 20
  - tcExamples.c, 36
- SLASH
  - tcutils.h, 39
- specie
  - tc\_species\_strings, 18
- str8
  - example3.c, 20
  - tcExamples.c, 36
- StringFuncString
  - libtc.h, 22
- strvect
  - example3.c, 20
  - tcExamples.c, 36
- TC\_API\_LIBRARY\_NAME
  - tcMain.c, 38
- TC\_BOOL
  - tc\_data\_defs.h, 28
- TC\_EPS
  - tc\_data\_defs.h, 26
- TC\_FLOAT
  - tc\_data\_defs.h, 29
- TC\_IARR
  - tc\_data\_defs.h, 29
- TC\_INT
  - tc\_data\_defs.h, 29

- TC\_LABEL\_STRING
  - tc\_data\_defs.h, 29
- TC\_MAX\_NR\_OF\_AXES
  - tc\_data\_defs.h, 26
- TC\_MAX\_NR\_OF\_CONST\_PER\_SUBLATTICE
  - tc\_data\_defs.h, 26
- TC\_MAX\_NR\_OF\_CONST\_PER\_SUBLATTICE\_IN\_IDEAL\_GAS
  - tc\_data\_defs.h, 26
- TC\_MAX\_NR\_OF\_CONSTITUENTS
  - tc\_data\_defs.h, 26
- TC\_MAX\_NR\_OF\_DATABASES
  - tc\_data\_defs.h, 26
- TC\_MAX\_NR\_OF\_ELEMENTS
  - tc\_data\_defs.h, 26
- TC\_MAX\_NR\_OF\_PHASES
  - tc\_data\_defs.h, 26
- TC\_MAX\_NR\_OF\_SPECIES
  - tc\_data\_defs.h, 26
- TC\_MAX\_NR\_OF\_SUBLATTICES
  - tc\_data\_defs.h, 27
- TC\_NWSE
  - tc\_data\_defs.h, 27
- TC\_NWSG
  - tc\_data\_defs.h, 27
- TC\_STRING
  - tc\_data\_defs.h, 29
- TC\_STRING\_LENGTH
  - tc\_data\_defs.h, 29
- TC\_STRLEN\_COMPONENTS
  - tc\_data\_defs.h, 27
- TC\_STRLEN\_CONSTITUENTS
  - tc\_data\_defs.h, 27
- TC\_STRLEN\_DATABASE
  - tc\_data\_defs.h, 27
- TC\_STRLEN\_ELEMENTS
  - tc\_data\_defs.h, 27
- TC\_STRLEN\_MAX
  - tc\_data\_defs.h, 27
- TC\_STRLEN\_PATH\_MAX
  - tc\_data\_defs.h, 27
- TC\_STRLEN\_PHASES
  - tc\_data\_defs.h, 27
- TC\_STRLEN\_REFERENCE
  - tc\_data\_defs.h, 27
- TC\_STRLEN\_SPECIES
  - tc\_data\_defs.h, 27
- TC\_STRLEN\_STOICHIOMETRY
  - tc\_data\_defs.h, 28
- TC\_VARS
  - tc\_data\_defs.h, 28
- TCDEV\_HOME
  - tcutils.h, 39
- TCFuncExport
  - tc\_data\_defs.h, 28
- TCHANDLE
  - tc\_data\_defs.h, 28
- TCPATH
  - tcutils.h, 39
- TEMP
  - tcutils.h, 39
- tc\_append\_database
  - \_tc\_function\_library, 10
  - tcapi.h, 31
- tc\_check\_license
  - \_tc\_function\_library, 10
  - tcapi.h, 31
- tc\_component\_status
  - \_tc\_function\_library, 10
  - tcapi.h, 31
- tc\_components\_strings, 15
  - component, 15
  - tc\_data\_defs.h, 28
- tc\_compute\_equilibrium
  - \_tc\_function\_library, 10
  - tcapi.h, 31
- tc\_conditions\_as\_arrays\_of\_strings, 15
  - condition, 16
  - tc\_data\_defs.h, 28
- tc\_constituents\_strings, 16
  - constituent, 16
  - tc\_data\_defs.h, 28
- tc\_create\_new\_equilibrium
  - \_tc\_function\_library, 10
  - tcapi.h, 31
- tc\_data\_defs.h, 24
  - BOOL\_FUNC\_WIN, 25
  - DllExport, 25
  - FLOAT\_FUNC\_WIN, 25
  - false, 25
  - INTEGER\_FUNC, 26
  - INTEGER\_FUNC\_GNU, 26
  - INTEGER\_FUNC\_WIN, 26
  - pointer, 28
  - TC\_BOOL, 28
  - TC\_EPS, 26
  - TC\_FLOAT, 29
  - TC\_IARR, 29
  - TC\_INT, 29
  - TC\_LABEL\_STRING, 29
  - TC\_MAX\_NR\_OF\_AXES, 26
  - TC\_MAX\_NR\_OF\_CONST\_PER\_SUBLATTICE, 26
  - TC\_MAX\_NR\_OF\_CONST\_PER\_SUBLATTICE\_IN\_IDEAL\_GAS, 26
  - TC\_MAX\_NR\_OF\_CONSTITUENTS, 26
  - TC\_MAX\_NR\_OF\_DATABASES, 26
  - TC\_MAX\_NR\_OF\_ELEMENTS, 26
  - TC\_MAX\_NR\_OF\_PHASES, 26
  - TC\_MAX\_NR\_OF\_SPECIES, 26
  - TC\_MAX\_NR\_OF\_SUBLATTICES, 27
  - TC\_NWSE, 27
  - TC\_NWSG, 27
  - TC\_STRING, 29
  - TC\_STRING\_LENGTH, 29
  - TC\_STRLEN\_COMPONENTS, 27



- TC\_STRLLEN\_CONSTITUENTS, 27
- TC\_STRLLEN\_DATABASE, 27
- TC\_STRLLEN\_ELEMENTS, 27
- TC\_STRLLEN\_MAX, 27
- TC\_STRLLEN\_PATH\_MAX, 27
- TC\_STRLLEN\_PHASES, 27
- TC\_STRLLEN\_REFERENCE, 27
- TC\_STRLLEN\_SPECIES, 27
- TC\_STRLLEN\_STOICHIOMETRY, 28
- TC\_VARS, 28
- TCFuncExport, 28
- TCHANDLE, 28
- tc\_components\_strings, 28
- tc\_conditions\_as\_arrays\_of\_strings, 28
- tc\_constituents\_strings, 28
- tc\_databases\_strings, 28
- tc\_elements\_strings, 29
- tc\_phases\_strings, 29
- tc\_reference\_strings, 29
- tc\_species\_strings, 29
- true, 28
- VOID\_FUNC\_WIN, 28
- tc\_database
  - \_tc\_function\_library, 10
  - tcapi.h, 31
- tc\_databases\_strings, 16
  - database, 16
  - tc\_data\_defs.h, 28
- tc\_define\_components
  - \_tc\_function\_library, 11
  - tcapi.h, 31
- tc\_degrees\_of\_freedom
  - \_tc\_function\_library, 11
  - tcapi.h, 31
- tc\_deinit
  - \_tc\_function\_library, 11
  - tcapi.h, 31
- tc\_delete\_condition
  - \_tc\_function\_library, 11
  - tcapi.h, 31
- tc\_delete\_symbol
  - \_tc\_function\_library, 11
  - tcapi.h, 31
- tc\_element
  - \_tc\_function\_library, 11
  - tcapi.h, 32
- tc\_element\_reject
  - \_tc\_function\_library, 11
  - tcapi.h, 32
- tc\_element\_select
  - \_tc\_function\_library, 11
  - tcapi.h, 32
- tc\_elements\_strings, 17
  - element, 17
  - tc\_data\_defs.h, 29
- tc\_enter\_ges5\_parameter
  - \_tc\_function\_library, 11
  - tcapi.h, 32
- tc\_enter\_symbol
  - \_tc\_function\_library, 11
  - tcapi.h, 32
- tc\_error
  - \_tc\_function\_library, 11
  - tcapi.h, 32
- tc\_function\_library
  - libtc.h, 22
- tc\_ges5
  - \_tc\_function\_library, 11
  - tcapi.h, 32
- tc\_get\_data
  - \_tc\_function\_library, 12
  - tcapi.h, 32
- tc\_get\_derivatives
  - \_tc\_function\_library, 12
  - tcapi.h, 32
- tc\_get\_ges5\_parameter
  - \_tc\_function\_library, 12
  - tcapi.h, 32
- tc\_get\_value
  - \_tc\_function\_library, 12
  - tcapi.h, 32
- tc\_init\_root
  - \_tc\_function\_library, 12
  - tcapi.h, 33
- tc\_init\_root3
  - \_tc\_function\_library, 12
  - tcapi.h, 33
- tc\_list\_component
  - \_tc\_function\_library, 12
  - tcapi.h, 33
- tc\_list\_conditions
  - \_tc\_function\_library, 12
  - tcapi.h, 33
- tc\_list\_phase
  - \_tc\_function\_library, 12
  - tcapi.h, 33
- tc\_list\_species
  - \_tc\_function\_library, 12
  - tcapi.h, 33
- tc\_list\_symbols
  - \_tc\_function\_library, 12
  - tcapi.h, 33
- tc\_nr\_of\_constituents\_in\_phase
  - \_tc\_function\_library, 12
  - tcapi.h, 33
- tc\_open\_database
  - \_tc\_function\_library, 13
  - tcapi.h, 33
- tc\_phase
  - \_tc\_function\_library, 13
  - tcapi.h, 33
- tc\_phase\_all\_constituents
  - \_tc\_function\_library, 13
  - tcapi.h, 33
- tc\_phase\_constituents
  - \_tc\_function\_library, 13

- tcapi.h, 34
- tc\_phase\_reject
  - \_tc\_function\_library, 13
  - tcapi.h, 34
- tc\_phase\_select
  - \_tc\_function\_library, 13
  - tcapi.h, 34
- tc\_phase\_status
  - \_tc\_function\_library, 13
  - tcapi.h, 34
- tc\_phase\_structure
  - \_tc\_function\_library, 13
  - tcapi.h, 34
- tc\_phases\_strings, 17
  - phase, 17
  - tc\_data\_defs.h, 29
- tc\_poly3
  - \_tc\_function\_library, 13
  - tcapi.h, 34
- tc\_put\_sitefractions
  - \_tc\_function\_library, 13
  - tcapi.h, 34
- tc\_read\_poly3\_file
  - \_tc\_function\_library, 13
  - tcapi.h, 34
- tc\_reference\_strings, 17
  - reference, 18
  - tc\_data\_defs.h, 29
- tc\_reject\_constituent
  - \_tc\_function\_library, 13
  - tcapi.h, 34
- tc\_reset\_error
  - \_tc\_function\_library, 14
  - tcapi.h, 34
- tc\_restore\_constituent
  - \_tc\_function\_library, 14
  - tcapi.h, 34
- tc\_save\_poly3\_file
  - \_tc\_function\_library, 14
  - tcapi.h, 35
- tc\_select\_equilibrium
  - \_tc\_function\_library, 14
  - tcapi.h, 35
- tc\_set\_component\_status
  - \_tc\_function\_library, 14
  - tcapi.h, 35
- tc\_set\_condition
  - \_tc\_function\_library, 14
  - tcapi.h, 35
- tc\_set\_license\_code
  - \_tc\_function\_library, 14
  - tcapi.h, 35
- tc\_set\_minimization\_option
  - \_tc\_function\_library, 14
  - tcapi.h, 35
- tc\_set\_phase\_addition
  - \_tc\_function\_library, 14
  - tcapi.h, 35
- tc\_set\_phase\_status
  - \_tc\_function\_library, 14
  - tcapi.h, 35
- tc\_set\_start\_value
  - \_tc\_function\_library, 14
  - tcapi.h, 35
- tc\_species\_status
  - \_tc\_function\_library, 14
  - tcapi.h, 35
- tc\_species\_strings, 18
  - specie, 18
  - tc\_data\_defs.h, 29
- tc\_version
  - \_tc\_function\_library, 15
  - tcapi.h, 35
- tcExamples.c, 36
  - example1, 36
  - example2, 36
  - example3, 36
  - ivect, 36
  - rvect, 36
  - str8, 36
  - strvect, 36
- tcExamples.h, 37
  - example1, 37
  - example2, 37
  - example3, 37
- tcMain.c, 37
  - importLibThermoCalc, 38
  - loadTCLibraryInCurrentDir, 38
  - main, 38
  - TC\_API\_LIBRARY\_NAME, 38
- tcapi.h, 29
  - tc\_append\_database, 31
  - tc\_check\_license, 31
  - tc\_component\_status, 31
  - tc\_compute\_equilibrium, 31
  - tc\_create\_new\_equilibrium, 31
  - tc\_database, 31
  - tc\_define\_components, 31
  - tc\_degrees\_of\_freedom, 31
  - tc\_deinit, 31
  - tc\_delete\_condition, 31
  - tc\_delete\_symbol, 31
  - tc\_element, 32
  - tc\_element\_reject, 32
  - tc\_element\_select, 32
  - tc\_enter\_ges5\_parameter, 32
  - tc\_enter\_symbol, 32
  - tc\_error, 32
  - tc\_ges5, 32
  - tc\_get\_data, 32
  - tc\_get\_derivatives, 32
  - tc\_get\_ges5\_parameter, 32
  - tc\_get\_value, 32
  - tc\_init\_root, 33
  - tc\_init\_root3, 33
  - tc\_list\_component, 33

- tc\_list\_conditions, 33
- tc\_list\_phase, 33
- tc\_list\_species, 33
- tc\_list\_symbols, 33
- tc\_nr\_of\_constituents\_in\_phase, 33
- tc\_open\_database, 33
- tc\_phase, 33
- tc\_phase\_all\_constituents, 33
- tc\_phase\_constituents, 34
- tc\_phase\_reject, 34
- tc\_phase\_select, 34
- tc\_phase\_status, 34
- tc\_phase\_structure, 34
- tc\_poly3, 34
- tc\_put\_sitefractions, 34
- tc\_read\_poly3\_file, 34
- tc\_reject\_constituent, 34
- tc\_reset\_error, 34
- tc\_restore\_constituent, 34
- tc\_save\_poly3\_file, 35
- tc\_select\_equilibrium, 35
- tc\_set\_component\_status, 35
- tc\_set\_condition, 35
- tc\_set\_license\_code, 35
- tc\_set\_minimization\_option, 35
- tc\_set\_phase\_addition, 35
- tc\_set\_phase\_status, 35
- tc\_set\_start\_value, 35
- tc\_species\_status, 35
- tc\_version, 35
- tclloadfunc
  - libtc.c, 20
- tcutils.c, 38
  - getTempEnvironmentPath, 38
  - getThermoCalcEnvironmentPath, 38
- tcutils.h, 39
  - getCurrentWorkingDir, 39
  - getTempEnvironmentPath, 39
  - getThermoCalcEnvironmentPath, 39
  - SLASH, 39
  - TCDEV\_HOME, 39
  - TCPATH, 39
  - TEMP, 39
- true
  - tc\_data\_defs.h, 28
- VOID\_FUNC\_WIN
  - tc\_data\_defs.h, 28
- VoidFuncInt
  - libtc.h, 23
- VoidFuncIntPIntPIntPIntP
  - libtc.h, 23
- VoidFuncNoParams
  - libtc.h, 23
- VoidFuncString
  - libtc.h, 23
- VoidFuncStringFloat
  - libtc.h, 23
- VoidFuncStringFloatP
  - libtc.h, 23
- VoidFuncStringFloatPFloatP
  - libtc.h, 23
- VoidFuncStringInt
  - libtc.h, 23
- VoidFuncStringIntInt
  - libtc.h, 23
- VoidFuncStringIntString
  - libtc.h, 23
- VoidFuncStringString
  - libtc.h, 23
- VoidFuncStringStringFloat
  - libtc.h, 23
- VoidFuncStringStringInt
  - libtc.h, 23
- VoidFuncStringStringIntIntFloatString
  - libtc.h, 24