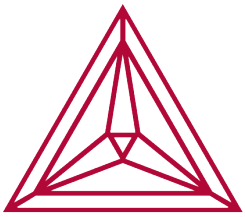
TC-API: SDK Programmer's Guide

Thermo-Calc Version 2019a



Thermo-Calc Software

Thermo-Calc Software AB

Råsundavägen 18, SE-169 67 Solna, Sweden

+46 8 545 959 30

documentation@thermocalc.com

www.thermocalc.com

TC-API

7.8.11410

Generated by Doxygen 1.8.9.1

Fri Sep 29 2017 19:50:12

Contents

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 thermodynmic 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 thermodynmic calculations to demonstrate the use of this api.
tcExamples.c
tcExamples.h

Main program to exemplify hos 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:

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

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 * tcloadfunc (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* tcloadfunc (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(* VoidFuncIntPIntPIntPIntP) (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(* VoidFuncIntPIntPIntPIntP) (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 DllExport __declspec(dllexport)
- #define TCFuncExport extern DllExport
- #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 DllExport __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 DllExport

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_↩ 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 tcpath)
- 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_STRIN↩ G element_array, TC_INT strLenElem, TC_FLOAT *number_of_sites)
- TC_INT tc_phase_constituents (TC_STRING phase_name, TC_INT *constituent_array, TC_STRIN↩ G 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 *linelen*)

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 "T↩ ABLE", "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 "sublattiice" 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 fallback

Definition at line 12 of file tcutils.c.

Index