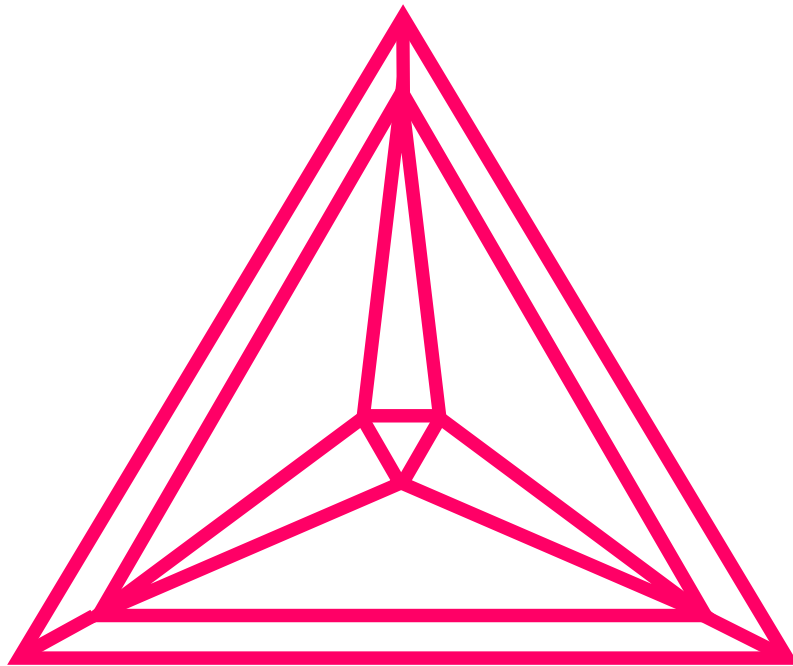


Thermo-Calc

Database Manager's Guide



Copyright © 1995-2014 Foundation of Computational Thermodynamics
Stockholm, Sweden

Contents

1	INTRODUCTION	2
1.1	THERMODYNAMIC AND KINETIC/MOBILITY DATABASES	2
1.3	ABOUT THIS DOCUMENT	2
1.4	THE DATABASE CHECKER PROGRAM	2
2	INITIALIZATION OF THE DATA MODULE	3
3	DATABASE DEFINITION FILE SYNTAX	5
3.1	ELEMENT	5
3.2	SPECIES	6
3.3	PHASE	6
3.4	CONSTITUENT	9
3.5	ADD_CONSTITUENT	9
3.6	COMPOUND_PHASE	10
3.7	ALLOTROPIC_PHASE	10
3.8	TEMPERATURE_LIMITS	10
3.9	DEFINE_SYSTEM_DEFAULT	10
3.10	DEFAULT_COMMAND	10
3.11	DATABASE_INFORMATION	11
3.12	TYPE_DEFINITION	11
3.13	FTP_FILE	12
3.14	FUNCTION	13
3.15	PARAMETER	14
3.16	OPTIONS	18
3.17	TABLE	18
3.18	ASSESSED_SYSTEMS	19
3.19	REFERENCE_FILE	21
3.20	LIST_OF_REFERENCE	21
3.21	ADD_REFERENCE	22
3.22	CASE AND ENDCASE	23
3.23	VERSION_DATA	23
4	DICTRA EXTENSIONS TO DATABASE DEFINITION FILE SYNTAX	24
4.1	PARAMETER	24
4.2	DIFFUSION	24
4.3	ZERO_VOLUME_SPECIES	25
5	EXAMPLES OF DATABASE DEFINITION FILES	26
5.1	EXAMPLE 1: A SMALL STEEL DATABASE	26
5.2	EXAMPLE 2: A PERSONAL DATABASE FOR THE SB-SN BINARY SYSTEM	27
5.3	EXAMPLE 3: A PUBLIC DATABASE FOR THE FE-CR-C TERNARY SYSTEM	29

1 Introduction

1.1 Thermodynamic and Kinetic/Mobility Databases

The Thermo-Calc Databases are created and maintained in the so-called Thermo-Calc Database Format (TDB) which has been used worldwide as an international standard for the CALPHAD-type thermodynamic calculations and kinetic simulations. This document gives a full description of the Thermo-Calc Database Format (with the DICTRA database extension), and details a comprehensive guide on how to properly create and manage a Thermo-Calc (and DICTRA) database.

Usually, a Thermo-Calc or DICTRA database is constructed as a single textual file with the extension of TDB, and the database name is normally given as *e.g.* PAQS2setup.TDB. But for some databases (especially when they are extremely large and when they are supposed to be used together with older versions of the Thermo-Calc or DICTRA software), one database may consist of one primary-definition file (`***setup.TDB`) and several other additional-definition TDB files (such as `***param.TDB`, `***funct.TDB`, `***refer.TDB`, *etc.*), which are built specifically for various defined parameters, functions, references and so forth, and which are linked to the `***setup.TDB` file through the proper usage of the `TYPE_DEFINITION` and/or `FTP_FILE` commands inside the `***setup.TDB` file; for details, refer to the *Sections 3.12* and *3.13* in this document.

However, with the most-current versions of the Thermo-Calc, TC-PRISMA and Thermo-Calc Software Development Kits (SDKs), you can effectively code everything into one single `***setup.TDB` file for a database at any scale.

1.3 About This Document

This document describes how a thermodynamic database or a kinetic database can be defined and constructed. This guide provides all kinds of details regarding not only the data structures and formats of the Thermo-Calc databases but also the DICTRA database-extensions. It helps the users to understand a database, to maintain a database, or even to construct a database. *Chapter 2* is on the initialization of the DATA Module in different computer environments. *Chapter 3* is about the Database Definition File syntax, which is mainly for thermodynamic database file while some of keywords are also valid for kinetic database file. The DICTRA extensions to the Database Definition File syntax are given in *Chapter 4*.

1.4 The Database Checker program

The Thermo-Calc software package includes a program for checking that the syntax of unencrypted Thermo-Calc or DICTRA database files is correct. The program applies the syntax rules that are set out in this document, and reports encountered errors and issues warnings. However, note that Thermo-Calc and DICTRA accept some deviations from these syntax rules. This means that a database may work even if the Database Checker reports errors and warnings. For example, an error is reported if an abbreviated phase name is found, but phase name abbreviations are accepted by Thermo-Calc and DICTRA.

The executable DatabaseChecker file is found in the Thermo-Calc home directory. The program can also be launched by selecting "Database Checker" from the "Tools" main menu in Thermo-Calc.

2 Initialization of the DATA Module

The following paragraphs will present how a *database initiation file* (or called *database directory file*) is constructed to work with the DATA module. Upon starting, DATA looks for this special file that has information on the available predefined databases (whether freely distributed with the Thermo-Calc/DICTRA packages, purchased from TCS or its agents, or created by the users).

The database initiation file is currently called TC_INITD.TDB (or TC_INITD) on MS-Windows, or initd.tdb on Linux/UNIX. Each user could have his own TC_INITD.TDB (or initd.tdb) file, but a better way is to define an environment variable with the name TC_INITD.TDB (or initd.tdb) translating to an initiation file name that is common for all users at one installation. This common file is automatically copied by the Thermo-Calc/DICTRA installation script to the \DATA\ area that is under the directory defined by the TCPATH parameter (on Windows), or the /data/ area that is under the directory defined by the TC_DATA parameter (on Linux/UNIX). The database manager can find it on a local computer for an independent installation locates or a connected server for a server installation.

When editing or modifying the definitions of initiation parameter (including short-names, paths and subdirectories, database definition file names, and instructive database descriptions) for the available databases in the TC_INITD.TDB or initd.tdb file, a specific format should be used, with the three sequent fields as explained below:

- ◆ The *first* field gives the abbreviated name for the database (currently restricted to maximum 5 characters)
- ◆ The *second* field specifies a path & filename (maximum 78 characters), containing the database definition and the actual database definition (setup) file name. The setup file name must consist of a main part and an extension of maximum 3 characters under Windows (such as SSOL5SETUP.TDB) or various Linux/UNIX platforms (such as ssol5setup.tdb). For the database path definition, the backslash “\” is used with MS-Windows, and the forward-slash “/” is used with Linux/UNIX. All databases should be normally located in subdirectories that are under the directory defined by the TCPATH parameter (under MS-Windows) or by the TC_DATA parameter (under Linux/UNIX), or under its subdirectory \DATA\ or /data/. Normally, the extension for a database definition (setup) file is either TDB (which stands for original textual TDB file) or TDC (which has been encrypted from the original TDB file).
- ◆ The *third* field (maximum 60 characters) details an optional full name (and version/subversion numbers) for the brief database description.

Each entry record (*i.e.*, for one single database) must end with an exclamation mark “!”. There should be no spaces or commas within the first and second fields since such characters are taken as field separators, while spaces or commons can be used within the third field.

Although that it is always OK to write each database entry in one or two lines (till its ending mark, !), it is usually recommended that the third field may be added on the same line as the first and second fields for MS-Windows environments, or on the following line for all kinds of Linux/UNIX platforms. The following is an example of an initiation file defining three databases named PURE5, SSOL5 and TCFE7:

Under MS-Windows environments:

```
SSOL5 TCPATH\DATA\SSOL5\SSOL5SETUP.TDC  SGTE Solutions Database version 5.1 !
PURE5 TCPATH\DATA\PURE5\PURE5SETUP.TDB  SGTE Pure Elements Database version 5.2 !
TCFE7 TCPATH\DATA\TCFE7\TCFE7SETUP.TDC  TCS Steels/Fe-Alloys Database version 7.0 !
```

Under Linux/UNIX platforms:

```
ssol5 TC_DATA/data/ssol5/ssol5setup.tdc
      SGTE Solutions Database version 5.1 !
pure5 TC_DATA/data/pure5/pure4setup.tdb
      SGTE Pure Elements Database version 5.2 !
tcfe7 TC_DATA/data/tcfe7/tcfe7setup.tdc
      TCS Steels/Fe-Alloys Database version 7.0 !
```

A DICTRA installation normally shares many directories with Thermo-Calc, including the \DATA\ (or /data/) area and the database initiation file TC_INITD.TDB (or initd.tdb).

For your convenience when switching/appending your self-generated databases/data-sets or lately-purchased databases from TCS or its agents, you can simply add them to the predefined database list in the database initiation file TC_INITD.TDB or initd.tdb of your installed Thermo-Calc/DICTRA packages; please also refer to Section 2.2.2.2 in the document of *Thermo-Calc Console Mode User's Guide*.

It is possible to write some comment lines, always starting with a "\$" sign, in the database initiation file; such lines are ignored by the DATA module. This is also applicable if the database manager or user wants to temporarily disable any database from the predefined database list for common uses.

If there are too many databases in the \DATA\ area that the DATA module may not be able to properly handle, the \$ sign can also be used to temporarily comment out some of the uncommon databases. However, this will not happen for almost all

Since TCCM, the NEW_DICTORY_FILE command (see Section 5.4.17 in the document of *Thermo-Calc Console Mode User's Guide*) was implemented so that Thermo-Calc can use several additional database initiation files that define accessing paths for various databases located in different subdirectories under the directory defined by the TCPATH or TC_DATA parameter. If Thermo-Calc is used under a MS-Windows environment, such additional database initiation files can be located at any directory of any driver, on either a local computer or connected server. The NEW_DICTORY_FILE command will pop up an *Open* window to access a database initiation file if the file name or its path is not given on the same line of the NEW_DICTORY_FILE command, or if it is incomplete or incorrect, so that the path (in the **Look in** box) and database initiation file name (in the **File name** box) can be appropriately selected. However, if Thermo-Calc is run on a Linux/UNIX platform, these files must be located in the current working directory (where Thermo-Calc is started).

In an additional database initiation file, the first database entry may need to have the same path definition structure as in the ordinary database initiation file TC_INITD.TDB or initd.tdb. So it is advisable to simply copy the entry line(s) for some common databases from the original TC_INITD.TDB or initd.tdb file to such a database initiation file. The entries of the additional databases follow. Similarly to the standard databases predefined in the TC_INITD.TDB or initd.tdb file, all databases should be normally located in subdirectories that are under the directory defined by the TCPATH parameter (on MS-Windows) or by the TC_DATA parameter (on Linux/UNIX), or under its subdirectory \DATA\ or /data/.

Given below is an example of an additional database initiation file, named like MYINITD1.TDB, for uses under Windows environments:

```
$
$ DATABASES TCC (Additional TCC Databases)
PURE5 TCPATH\DATA\PURE5\PURE5SETU.TDB    SGTE Pure Elements Database, version 5 !
AD1   TCPATH\DATA\ADD1\AD1SETUP.TDB     TCS ADD1 Solution Database, version 1 !
AD2   TCPATH\ADDDATA\ADD2\AD2SETUP.TDB  TCS ADD2 Solution Database !
$AD2o TCPATH\ADDDATA\ADD2old\AD2SETUP.TDB TCS ADD2 Database (old) !
AD3   TCPATH\DATA\NEWDATA\ADD3\AD3SETUP.TDB TCS ADD3 Solution Database !
AD4   TCPATH\DATA\NEWDATA\MYPROJ1\ADD4\AD4SETUP.TDB
      MYPROJECT1 ADD4 Solution Database !
$
$ DATABASES DIC (Additional DICTRA Databases)
DCAD1 TCPATH\DICDATA\DCADD1\DCAD1SET.TDB
      TCS DCADD1 Mobility Database !
```

3 Database Definition File Syntax

The database definition (*.TDB) file (normally named as ***setup.TDB) consists of a set of keyword codes each followed by one or several parameters (arguments). A complete keyword entry must end with an exclamation mark "!". A single keyword entry can be up to 2000 characters long, but the maximum length of a line in an *.TDB file is 78 characters, so it may be necessary to continue the keyword's parameters (arguments) in several lines, and the exclamation mark "!" must be written at the end of the last line. It is recommended to *always have at least one empty space at the beginning of each continuation line for keyword's parameters (arguments)*; otherwise, the DATA module may misunderstand the parameters (or arguments), or issue some error messages when reading the keyword entry. The keyword and its various parameters (arguments) are separated by a space or a comma. A dollar sign "\$" in the first position of the line indicates that the line is a comment line, which is ignored by the DATA module.

The DATA module completely reads the database definition file (***setup.TDB) only once, from beginning to end, when the database is selected. Much checking is done by the DATA module when reading the definition file. This implies that (nearly) everything must be declared or defined before it is used in any other way. For example, if the GRAPHITE phase is to be included in the database definition, the element C (carbon) and the phase GRAPHITE must be defined before declaring that carbon dissolves in graphite. This definition order is necessary to build the internal data structure acceptable by the DATA module (during its consistency checking).

This section gives a description of the available keywords and their appropriate arguments. The reader is assumed to have basic knowledge of the Gibbs Energy System (GIBBS) module.

The following syntax is used:

KEYWORD [arg.1]*# [arg.2]*## {optional arg.3}!

The keywords are in the text written in full length but can be abbreviated as long as the abbreviation is unique. A keyword may have syntax consisting of several arguments and optional arguments. The number, # or ##, in the notation, [...] *# or [...] *##, indicates an argument with a maximum length of ASCII characters. Arguments within square brackets [...] *must* always be given, while such in curly braces { . . . } are *optional*.

3.1 ELEMENT

ELEMENT [element name]*2 [ref. state]*24 [mass] [H298] [S298] !

The **element name** (maximum 2 characters) is meant to be the one found in the periodic chart but there is no restriction on naming conventions. However, the GIBBS module only recognizes UPPER-case element names (if the Upper Case Mode has been selected by the GIBBS command REINITIATE), implying lower-cases (if defined in a database) is automatically converted to UPPER-cases by the DATA/GIBBS module. The elements are automatically entered as species using the same names of the elements. If, for instance, the species corresponding to FE by some reasons needs to be named FE1, it is possible to define the species as FE1, resulting in an element named FE and a species named FE1. Vacancies (VA) and electrons (denoted either as /- in gaseous, liquid or solid phases, or as ZE in an aqueous solution phase), if used, need to be also entered as special elements for correct handling by the DATA module.

The **reference state** (maximum 24 characters) is the stable phase (at 298.15 K and 1 bar) that should contain this element and should be used as the reference state for all thermodynamic data of this element. The **mass**, given in gram per mole, is used in various calculation programs and should always be given the correct value. **H298** and **S298** denote the enthalpy and entropy difference between 0 and 298.15 K for the element in SI units. If they are unknown, the values can be set to zero. All these information (**reference state**, **H298** and **S298**) precisely define the so-called **SER (Stable Element Reference State)**.

Examples:

```
ELEMENT /- ELECTRON_GAS      0.0  0.0  0.0  !
ELEMENT VA VACUUM           0.0  0.0  0.0  !
ELEMENT ZE UNIT_CHARGE      0.0000000001  0.0  0.0  !
ELEMENT AL FCC_A1           26.98154  4577.296  28.3215  !
ELEMENT C GRAPHITE          12.011  1054.0  5.74  !
ELEMENT FE BCC_A2           55.847  4489  27.28  !
ELEMENT O 1/2_MOLE_O2 (G)   15.9994  4341  102.5158  !
ELEMENT TI HCP_A3           47.88  4810  30.648  !
```

```
ELEMENT ZR HCP A3          91.224  5566.27  39.181 !
ELEMENT ZY DUMMY          1          1          1 !
```

3.2 SPECIES

SPECIES [*species name*]*24 [*stoichiometric formula*] !

This keyword defines species in the data structure. Every **species name** (maximum 24 characters) must be unique. The species are built from the already defined set of elements in the stoichiometric formula. If a non-defined element is referred, DATA will issue an error message and the data structure is probably damaged. The species names do not necessarily have to be the same as the stoichiometry formula, but it is often recommended to do so for most cases. The elements are automatically entered as species using the same names of the elements. One can define a species name as a mixture of UPPER-case and lower-case in a database, but the DATA module will automatically convert all lower-cases to UPPER-cases, as the GIBBS module only recognizes UPPER-case species names. If wished for some convenience in naming the species in a database, one can also use some special characters [such as +, -, _, / and .] in species names, but one should avoid using other special characters [such as (and)].

The **stoichiometric formula** is written with a simplified chemical notation, in which the chemical elements should always be given in UPPER-cases and in any preferred order, and their stoichiometric coefficients can be written in either real numerical factor or integer digits. It is important that the numerical factor of 1 cannot be left out. Subgroups are not allowed in a stoichiometry formula; however, if wished for some convenience in specifying the stoichiometry formula for a specific species in a database, one can specify it in a way that some elements (always together with their corresponding partial stoichiometric coefficients) can be repeated (as shown below in some examples).

Examples:

```
SPECIES AL2O3          AL2O3 !
SPECIES Silica        SI1O2 !
SPECIES NaSb_6OH      NA1SB1O6H6 !
SPECIES FE+2          FE/+2 !
SPECIES SB-3          SB/-3 !
SPECIES AlCl2/3        AL.33333CL.666667 !
SPECIES AL1CL1H2O2     AL1CL1H2O2 !
SPECIES AlCl3_3H2O     AL1CL3H6O3 !
SPECIES AlO2H2Cl.H6O3  AL1O2H2CL1H6O3 !
SPECIES AlCl2-OH.3H2O  AL1CL2O1H1H6O3 !
SPECIES AlCl2OH.3Water AL1O1H1CL2H6O3 !
```

3.3 PHASE

PHASE [*phase name*]*24 [*data-type code*]*8 [*numb. subl.*] [*sites in subl. 1*] [*sites in subl. 2*] etc... {*auxiliary text string*} !

This keyword defines a phase and its properties (except for what species are allowed to enter it and for its thermodynamic parameters).

The **phase name** (maximum 24 characters) must be unique; otherwise the DATA module sees it as an attempt to redefine a previously defined phase. This mistake causes DATA to issue an error message and ignore the rest of the line. A phase name can be suffixed by a “_” sign and some simple letters to identify the physical state(s) or structure type(s) of the phase; some examples of recommended suffixes are:

```
ABC_S      ⇒ The ABC phase in solid state
ABC_S2     ⇒ The ABC phase in solid state 2
ABC_S3     ⇒ The ABC phase in solid state 3
ABC_LT     ⇒ The ABC phase in solid state at low temperatures
ABC_HT     ⇒ The ABC phase in solid state at high temperatures
ABC_L      ⇒ The ABC phase in liquid state
ABC_LIQ    ⇒ The ABC phase in liquid state
FCC_A1     ⇒ The FCC phase in disordered structure type A1
FCC_L12    ⇒ The FCC phase in ordered structure type L12
```

The phase name can also be attached with a colon sign “:” and a letter for a legal GIBBS phase-type code (*e.g.*, IONIC_LIQ:Y and GAS:G); see more examples below.

Legal GIBBS phase-type codes are:

- G \Rightarrow Bit set for a gaseous mixture phase.
- A \Rightarrow Bit set for an aqueous solution phase.
- Y \Rightarrow Bit set for an ionic liquid solution phase (that is specially treated by the Ionic Two-Sublattice Liquid Model).
- L \Rightarrow Bit set for a liquid solution phase [but not A (aqueous) or Y (ionic liquid)].
- I \Rightarrow Bit set for a phase with charged species [but not G (gaseous), A (aqueous) or Y (ionic liquid)].
- F \Rightarrow Bit set for an ordered FCC or HCP solution phase with 4 substitutional sublattices (additionally, such a phase can also have an interstitial sublattice).
- B \Rightarrow Bit set for an ordered BCC solution phase with 4 substitutional sublattices (additionally, such a phase can also have an interstitial sublattice).

Note that other invalid characters (*e.g.* M or P) will be eventually treated, together with the colon sign “:”, as a part of a phase name.

A G phase (gaseous mixture) or an A phase (aqueous solution) is usually treated as a substitutional phase without sublattice, and that an L phase (ordinary liquid solution) is normally (but not always) modelled as a substitutional phase without sublattice, too.

The two phase-type F and B are useful for ordered FCC (or HCP) and BCC solution phases handled by the so-called *Four Substitutional-Sublattice Ordering Model*, which always require that the solution phase must have 4 sublattices for substitutional ordering and can additionally have an interstitial sublattice.

For ordered FCC or HCP phases, these four substitutional sublattices represent four corners of the regular tetrahedron on these lattices all of which are the nearest neighbours. A *Normal 4-Sublattice Model* requires that all the G parameters for each of end-members with the same elements but distributed on different sites be given separately. However, as these corners are identical lattice points, the phase-type option F means that the G parameters need be given only once. The possible permutations are handled automatically. *To be more clarified:* An A–B binary solution phase (with the element A locates on one sublattice site and B on the other three sublattice sites) treated by the Normal 4-Sublattice Model has to have 4 G parameters for 4 end-members, *i.e.*, G(phase, A:B:B:B), G(phase, B:A:B:B), G(phase, B:B:A:B) and G(phase, B:B:B:A), because of that in the general case these G parameters can be different from each other. But for the FCC and HCP orderings, they are identical and thus all G parameters of such end-members need be given only once, and the possible permutations are then automatically handled by the GIBBS module. This significantly simplifies the usage of this model (*Four Substitutional-Sublattice Ordering Model*) in multicomponent alloys.

There are restrictions on the excess parameters allowed with the phase-type option F. One can only have excess parameters of the following types:

- L(phase, A, B: *: *: *: *; 0...9)
- L(phase, A, B:C, D: *: *: *; 0...9)

The asterisk “*” means that the interaction parameter is independent of the constituents on its corresponding sublattice. No ternary interaction parameters (*i.e.*, with 3 elements on one sublattice site) are allowed. The reason for this restriction is that it would be too complicated to handle all possible permutations. In the current *Four Substitutional-Sublattice Ordering Model*, the binary interaction between A and B atoms is thus independent of the constituents on the other sublattices, where there are many other parameters to model the composition-dependence of the Gibbs energy (both in the ordered and disordered parts of the phase). The model for these ordered phases are always partitioned in a disordered part (with a single substitutional sublattice) and an ordered part (with 4 substitutional sublattices for ordering).

For ordered BCC phases, the phase-type option B means the same thing but the situation is a bit more complicated, as the 4-substitutional-sublattice ordering phase represents an irregular tetrahedron with 2 pairs of sites that are next nearest neighbours. Thus, for an A–B binary solution phase (with the element A locates on two sublattice site and B on two sublattice sites) treated by the Normal 4-Sublattice Model, the end-member described by the G(phase, A:A:B:B) term has 4 nearest neighbor bonds between A and B atoms, whereas the end-member described by the G(phase, A:B:A:B) term has 2 nearest neighbour bonds between A and B atoms and 2 next nearest neighbor bonds (for detailed reference one may visit the Crystal Lattice Structure web page <http://cst-www.nrl.navy.mil/lattice/index.html>).

The first end-member [described by the G(phase, A:A:B:B) term] represents B2-ordering and the second [described by the G(phase, A:B:A:B) term] stands for B32-ordering. There are 2 permutations of the G(phase, A:A:B:B) term and 4 permutations of the G(phase, A:B:A:B) term, automatically


```
PHASE FCC_L12:F %&AX 5 0.25 0.25 0.25 0.25 1.0
> Ordered FCC phase, modelled by 4-Sublattice Model for Ordering. !
PHASE AQUEOUS:A %HIJMR 1 1.0
> Aqueous Solution: using the Complete Revised HKF Model. !
```

3.4 CONSTITUENT

CONSTITUENT [phase name]*24 [constituent description]*2000 !

This keyword (and the ADD_CONSTITUENT keyword for large solution phase) defines the *phase-constitution* as a list of *constituents* (for a substitutional phase with no sublattice) or of *constituent arrays* (for a sublattice phase).

The **phase name** (maximum 24 characters) must be an already defined phase (*i.e.*, already through the PHASE keyword). Specifying the phase name always in UPPER-cases is recommended; however, if one would prefer to write it as a mixture of UPPER-case and lower-case in a database, the DATA module will then automatically convert all lower-cases to UPPER-cases, as the GIBBS module only recognizes UPPER-case phase names. It is important that if a phase bears a legal phase-type (among G, A, Y, L, I, F and B) in its phase definition (already by the PHASE keyword; such as GAS:G, SLAG:L, LIQUID:L, IONIC_LIQ:Y, SPINEL:I, FCC_L12:F, HCP_D021:F, BCC_B2:B, AQUEOUS:A), such a valid phase-type code must also always be attached to the phase name in the CONSTITUENT keyword (and the ADD_CONSTITUENT keyword).

The **constituent description** (maximum 2000 characters) is a list of the species that enter a phase. The list starts with a colon “:”, indicating the start of the sub-list of species for the first sublattice, and different sublattices are separated by colons. The complete sequence ends with a final colon. Optionally, each sublattice may specify which species are considered to be so-called major constituents. This is done by adding a “%” (percent sign) directly on the species name. The start values on the site fractions of the major constituents should sum to 0.99 on a specific sublattice. Thus, the so-called minor constituents (*i.e.*, those without a “%”) adds up to 0.01. Maximum 2000 characters can be coded in the constituent description, continuing in sequent lines. If the phase has a constituent description longer than 2000 characters, the rest can be coded in one or several ADD_CONSTITUENT keywords.

Examples:

```
CONSTITUENT BCC_A2 :FE
CONSTITUENT IONIC-LIQ:Y :FE+2 : SB-3: !
CONSTITUENT M23C6 :CR FE :FE CR W MO : C: !
CONSTITUENT AQUEOUS:A :H2O% AG+1 AGF AGCL AGCL2-1 AGI3-2 AGSO4-1 AGC2H4+1
  AGN2H6+1 AGC2N2-1 AGC2H4NO2 AL+3 ALF3 ALO2-1, ... : !
CONSTITUENT SPINEL:I : AL+3% CR+3 FE+2% FE+3 MG+2% NI+2
  : AL+3% CA+2 CR+3 FE+2 FE+3 MG+2% NI+2 VA
  : FE+2 MG+2 VA%
  : N-3 O-2% :!
```

3.5 ADD_CONSTITUENT

ADD_CONSTITUENT [phase name]*24 [constituent description]*2000 !

This keyword adds more constituents to a phase that has already some constituents. **Its syntax is the same as for the CONSTITUENT keyword.** This keyword can be used several times, if the phase is very large, *e.g.*, a gaseous mixture or a complex aqueous solution. This is useful when there are so many constituents in a phase that the 2000 characters available for the constituent description list are not enough.

Note that constituents are not necessary on all sublattices. In the second example below, no addition is made to the first sublattice.

Examples:

```
ADD_CONSTITUENT GAS :S1 S2 S3 ... : !
ADD_CONSTITUENT IM-PHASE : :CR:W ... : !
ADD_CONSTITUENT AQUEOUS:A :CUCL+1 CUCL2 CUCL2-1 CUCL3-2 CUOH+1 CUO2H2 CUO3H3-1
  CUO4H4-2 CU2OH+3 CU2O2H2+2 CU3O4H4+2 NIO2H2 NIO3H3-1 NIO4H4-2 NI2OH+3
  NI4O4H4+4 ZNOH+1 ZNO2H2 ZNO3H3-1 ZNO4H4-2 ... : !
```

3.6 COMPOUND_PHASE

COMPOUND_PHASE [phase name]*24 [data-type code]*8 [constituent] !

The keyword is a compact way to simultaneously define a species, a compound phase (maximum 24 characters) and its phase-constituent. It is useful for stoichiometric phases with constant compositions. The species name and stoichiometric formula must be identical, *i.e.*, being the given `constituent`. The phase will have this species as its only constituent. This keyword was implemented to make the database definition file for a large substance database more compact; it is just a combination of the `SPECIES`, `PHASE` and `CONSTITUENT` keywords.

Examples:

```
COMPOUND_PHASE AL2O3 % AL2O3 !
COMPOUND_PHASE MAGNETITE %MF FE3O4 !
COMPOUND_PHASE QUARTZ % SIO2 !
```

3.7 ALLOTROPIC_PHASE

ALLOTROPIC_PHASE [phase name]*24 [data-type code]*8 [constituent] !

This keyword does the same as the `COMPOUND_PHASE` keyword for entering an allotropic phase (maximum 24 characters), but does not enter the constituent as a species to the data structure; it should therefore be used if the species has already been defined.

Examples:

```
ALLOTROPIC_PHASE BETHA-AL2O3 % AL2O3 !
ALLOTROPIC_PHASE CRISTOBALITE % SIO2 !
ALLOTROPIC_PHASE TRIDYMITTE % SIO2 !
```

3.8 TEMPERATURE_LIMITS

TEMPERATURE_LIMITS [lower limit] [upper limit] !

This keyword sets the default upper and lower temperature limits used by the GIBBS module for Gibbs energy parameters and functions. It can be used only once in one database definition file and all its sequential files.

Example:

```
TEMPERATURE_LIMITS 500.0 1800.0 !
```

3.9 DEFINE_SYSTEM_DEFAULT

DEFINE_SYSTEM_DEFAULT [keyword] {G-ref. type index} !

This keyword sets the default value to `ELEMENT` or `SPECIES` in the `DATA` command `DEFINE_SYSTEM` (see [Section 5.4.7 in the document of *Thermo-Calc \(Console Mode, Version 3.0\) User's Guide*](#)). For a substance database, it might be appropriate to have `ELEMENT` as default value whereas a large solution database can benefit from having `SPECIES` as default value. A proper default value is beneficial for a beginner, but an advanced user will probably use the `DATA` commands `DEFINE_ELEMENT` and `DEFINE_SPECIES` to override the default value.

{**G-ref. type index**} is an integer indicating the reference state type for an element when entering and listing data in the GIBBS module. The following lists legal numbers and their corresponding meaning (the reference state type for an element):

- 1 ⇒ symbol: G
- 2 ⇒ symbol: H298
- 3 ⇒ symbol: H0

Example:

```
DEFINE_SYSTEM_DEFAULT element 2 !
```

3.10 DEFAULT_COMMAND

DEFAULT_COMMAND [secondary keyword and parameters] !

The keyword specifies commands to be executed by the DATA module at database initialization. The syntax of the available commands is currently not the same as the user available DATA commands but the actions are similar. The available secondary keywords and parameters in syntax for DEFAULT_COMMAND are,

```

DEFINE_SYSTEM_ELEMENT [element names]
DEFINE_SYSTEM_SPECIES [species names]
DEFINE_SYSTEM_CONSTITUENT [phase] [sublattice] [species]
REJECT_SYSTEM_ELEMENT [element names]
REJECT_SYSTEM_SPECIES [species names]
REJECT_SYSTEM_CONSTITUENT [phase] [sublattice] [species]
REJECT_PHASE [phase names]
RESTORE_PHASE [phase names]

```

Examples:

```

DEFAULT_COMMAND DEFINE_SYSTEM_ELEMENT FE VA !
DEFAULT_COMMAND REJECT_SYSTEM_CONSTITUENT LIQUID 2 C !
DEFAULT_COMMAND REJECT_PHASE LIQUID !
DEFAULT_COMMAND RESTOR_PHASE GAS !

```

3.11 DATABASE_INFORMATION

DATABASE_INFORMATION [text]*10000 !

This keyword defines a text for the detailed description of the current database. The text can be listed with the DATA command DATABASE_INFORMATION. An apostrophe, “'”, can be used in the text to indicate a new line; and two apostrophe, “''”, can be used in the text to indicate a new line plus an empty line. Note that the continuous text length (each line with max 78 characters) is 10,000 characters

Example:

```

DATABASE_INFORMATION This is the XXX-Alloy Solution Database '
    in the A-B-C-D-..... System. '
    Developed by TCS, released in May 2001. ''
... more ... !

```

3.12 TYPE_DEFINITION

TYPE_DEFINITION [data-type code]*1 [secondary keyword with parameters] !

This keyword couples phases to an action performed by the DATA module when the DATA command GET_DATA is executed. The available secondary keywords and associated parameters in syntax for TYPE_DEFINITION are:

```

SEQ [filename]
RND# [filename]
GES [valid GIBBS command with parameters]
POLY3 [valid POLY command with parameters]
TDB [valid DATA command with parameters]
IF [conditional statement] THEN [keyword with parameters]
AFTER [valid GIBBS command with parameters]

```

The secondary keyword **SEQ** specifies a sequential file that stores parameters belonging to the phases using the associated data type code (which is defined by this TYPE_DEFINITION keyword). A special case where the filename is given as an asterisk, *, implies that the database definition file also acts as a sequential data storage file. This case makes it possible to have a single file for a small database, which is especially suited for personal databases.

The secondary keyword **RND** should be concatenated with a positive integer # to indicate the type of the *random file*. Currently, there are three types of random files. Type RND0, the default type, is used for complete Gibbs energy expressions (G0 parameters), where the search field is the unabbreviated parameter name. Type RND1 is designated for functions, where the function name is used as the search field. Type RND2 is reserved for binary interaction parameters, where its search field is also the unabbreviated parameter name without any interaction order notation. Note that ternary and higher order interaction parameters must be specified on a sequential file. Moreover, the internal structures of these random files are subject to changes with different versions of DATA, and with implementations of DATA on various computer systems. For further information, one may consult the FORTRAN program **TDBSORT**, which is available from Thermo-Calc Software AB.

The secondary keyword **GES**, **POLY3**, or **TDB** specifies a modification of, or addition to, phases having the associated data type code, such as magnetic contribution, another excess model, or any other valid GIBBS/POLY/DATA command that applies to a certain phase. By implementing this as a call to the interactive GIBBS/POLY/DATA module, much flexibility is achieved. If a new type of addition is implemented in GIBBS/POLY/DATA module, it can immediately be used in the database definition file without reprogramming the DATA module. Note, in several examples given below, the use of a "@" (at) sign indicates any phase to which the relevant type definition (*e.g.*, A, B, 4, or E) applies.

The secondary keyword **IF/THEN** allows specification of a conditional statement structured with respect to the phase constitution that controls the execution of a following type-definition [keyword with parameters] string. See the last four examples below.

The secondary keyword **AFTER** is similar to the GES keyword except the defined GIBBS command will be executed after all parameters have been entered. The reason for this is that the command has no effect unless there is a parameter. After "AFTER" keyword a GIBBS command must be given and it is executed AFTER entering the parameters of the phase.

The **data-type** code (always as one string) can be any normal or special character, *e.g.*, 0, 5, A, F, M, %, &, (, *etc.*, and is referred in the definition keywords (PHASE, COMPOUND_PHASE and ALLOTROPIC_PHASE) for various phases.

A phase can have several so-called ADDITIONAL parts of different types (that are enforced by certain TYPE_DEFINITIONS which call the GIBBS command AMEND_PHASE_DESCRIPTION for describing various contributions to Gibbs energy); however, the DATA/GIBBS modules will automatically delete any early-defined ADDITIONAL part(s) of the same type.

The DATA module can selectively retrieve functions which are necessary for a defined system from a database that has functions stored in its setup file or SEQ sequential function file, while all other functions irrelevant for the defined system are simply ignored and will not be saved in associated GIBBS and POLY workspaces.

Examples:

```
TYPE_DEF % SEQ TCPATH\DATA\[DATABASE]\PARAMETERS.TDB !
TYPE_DEF I SEQ TCPATH\DATA\[DATABASE]\INTERACTION-PARAMS.TDB !
TYPE_DEF G RND0 TCPATH\DATA\[DATABASE]\GZERO-PARAMS.TDB !
TYPE_DEF F RND1 TCPATH\DATA\[DATABASE]\FUNCTIONS.TDB !
TYPE_DEF & RND2 TCPATH\DATA\[DATABASE]\BINARY-INTERACTIONS.TDB !
TYPE_DEF A GES AM_PH_DES @ MAGNETIC -1 0.40 !
TYPE_DEF B GES AM_PH_DES @ MAGNETIC -3 0.28 !
TYPE_DEF 4 GES AM_PH_DES @ EXCESS_MODEL REDLICH-KISTER-KOHLER !
TYPE_DEF 5 GES AM_PH_DES AQUEOUS EXCESS_MODEL HKF !
TYPE_DEF 6 GES AM_PH_DES AQUEOUS HKF ELECTROSTATIC !
TYPE_DEF 7 GES AM_PH_DES AQUEOUS STATUS 02084000,,, !
TYPE_DEF 8 GES AM_PH_DES AQUEOUS MAJOR_CONST 1 H2O !
TYPE_DEF E AFTER AM_PH_DES LIQUID EXCESS MIXED-EXCESS
      A B LEGENDRE C A POLYNOM ,,,, !
TYPE_DEF T AFTER AM_PH_DES LIQUID TERN-EXT TOOP-KOHLER B A C ,,,, !
TYPE_DEF Q AFTER AM_PH_DES LIQUID TERN-EXT KOHLER FE CR NI !
TYPE_DEF C IF (PD AND PT AND SN) THEN TDB RESTORE_PHASE BCT_A5 !
TYPE_DEF D IF (PD AND (PT OR SN)) THEN TDB REJECT_PHASE BCC_A2 !
TYPE_DEF E IF ((NB OR TI OR V) AND (C OR N)) THEN
      GES AM_PH_DES @ COMP_SET ,, CR NB TI V: C N: !
TYPE_DEF F IF (ALO3/2 OR CRO3/2 OR FEO OR MNO OR SIO2) THEN
      GES AM_PH_DES LIQUID COMP_SET ,, ALN%,ALO3/2%,CRO3/2%,
      FEO%,FEO3/2%,MNO3/2%,MNS%,SIO2%,TIO2% : !
TYPE_DEF R GES AM_PH_DES FE LIQUID FRACTION_LIMITS Fe 0 0.6
      Ag 0 0.01 Al 0 0.05 Ca 0 0.05 Co 0 0.01 Cr 0 0.01
      Cu 0 0.02 Mg 0 0.05 Mn 0 0.05 Mo 0 0.05 Nb 0 0.05
      Ni 0 0.05 Pb 0 0.05 Si 0 0.10 Sn 0 0.02 Ti 0 0.05
      U 0 0.01 V 0 0.02 W 0 0.02 Zr 0 0.03
      B 0 0.01 C 0 0.01 H 0 0.01 N 0 0.01 O 0 0.01
      P 0 0.01 S 0 0.01 !
```

3.13 FTP_FILE

FTP_FILE [filename] !

The FTP file is a special function random file whose function names correspond to the record numbers where these record names and their functions are stored. The FTP file dramatically decreases search time for the associated database in the DATA module. The file is used for large substance databases along with a SEQ sequential or RND0 random file for storage of G0 parameters referring the functions named **FxxxxT** that are stored in the FTP file. The integer number **xxxx** is a search code used by DATA when such files are processed. No modification of this file type is allowed.

Example:

```
FTP_FILE TCPATH\DATA\[DATABASE]\FTP-FILE.DATA !
```

3.14 FUNCTION

```
FUNCTION [function name]*8 [lowest temp. limit]
  [expression 2]; [upper temp. limit 2] Y
  [expression 1]; [upper temp. limit 1] Y
  [expression 3]; [upper temp. limit 2] Y
  ..... ; ..... Y
  [expression n-1]; [upper temp. limit n-1] Y
  [expression n]; [upper temp. limit n] N {Ref. Index} !
```

GIBBS has the capability to use predefined functions in the expression (TP-Function) of a Gibbs energy parameter or in other functions. This is often used when several parameters (or functions) have a common sub-expression, such as for metastable modifications of elements. The FUNCTION keyword can appear in both files for database definition and sequential storage, but not in FTP files. A valid function name can have up to 8 characters.

A function always starts with a lowest temperature limit of its applicability, followed by one or more (up to 10) expressions (TP-Functions) that are coded as mathematical relations of constants, functions of stable variables (T and P) and other entered functions (normally with a # suffix, e.g. +3*GHSERAL#). For further information, see Section 11.2.5 (Functions of Temperature and Pressure) in the *Thermo-Calc User's Guide* document. The expression is a FORTRAN-like expression and operators +, -, *, = and ** can be used (** only with integer powers). Unary-functions LN or LOG (both for natural logarithm) and EXP (for exponential) can also be used. Each expression (TP-Function) should end with a semicolon ";", and be followed by its upper applicable temperature limit and a continuation indicator (Y for continuing with the next expression, or N for ending the function's expression). If there is no continuation after a specific expression (TP-Function), the reference index can optionally be given after the N indicator.

A complete/valid function entry can be written in several continuation lines if the function's expression (TP-Function) is too long or if there are more than one applicable expression (TP-Function), as the maximum length of each line is 78 characters. However, it is recommended to **always have at least one empty space at the beginning of each continuation line**. Otherwise, the DATA module may misunderstand the expression or issue some error messages when reading the function entry. One should avoid entering functions like

```
FUNCTION GHSERXY 298.15
-1000+1058*T-38.9*T*LOG(T)+GFUNXY#; 6000 N !
```

Such a function will be read by the DATA module as 1000+1058*T-38.9*T*LOG(T)+GFUNXY#, rather than as

```
-1000+1058*T-38.9*T*LOG(T)+GFUNXY#.
```

The reason for this is that the DATA module will concatenate all lines and remove extra spaces before trying to enter the function in the GIBBS workspace. Thus, the - sign will be taken as delimiter between 298.15 and 1000, and the function will wrongly become

```
FUNCTION GHSERXY 298.15 1000+1058*T-38.9*T*LOG(T)+GFUNXY#; 6000 N !
```

This mistake can be avoided by giving at least one empty space as the first character of a new line, such as

```
FUNCTION GHSERXY 298.15
 -1000+1058*T-38.9*T*LOG(T)+GFUNXY#; 6000 N !
```

which will be correctly read as

```
FUNCTION GHSERXY 298.15 -1000+1058*T-38.9*T*LOG(T)+GFUNXY#; 6000 N !
```

The lowest-temperature limit (in Kelvin) for the applicability of the (first) TP-Function in a function is normally set by default as 298.15 K, in most cases; however, one may set another limit when it is applicable (according to experimental data and assessments). An upper-temperature limit (in Kelvin; followed by an Y or N sign) for the applicability of each TP-Function in a function must be given after the semicolon ";" sign immediately following the specific TP-Function; and the highest-temperature limit (in Kelvin) for the applicability of the current function is always followed by the N sign. If a negative number is given as the lowest-temperature limit, it will be assumed that there are breakpoints in pressure for this function; in such a case, it is interpreted as the lowest-pressure limit (in Pascal), and the other limits in the current function will also be taken as pressure limit values (in Pascal).

The temperature/pressure limits for the functions are checked during calculations. An indicator will be set if the actual temperature/pressure condition is below the lowest temperature/pressure limit or above the highest temperature/pressure limit; and in such a case, an extrapolation will be conducted using the TP-Function valid in the nearest temperature/pressure range.

Note that, in general, a function must be defined before it is referred. However, this rule may be neglected by suffixing a number sign "#" directly on the function name (such as +3*GHSERAL#). However, this has been further simplified since TCCN/TCW1, where functions no longer need to be suffixed with #. A function that is used in a parameter (or other functions) but has never defined in the currently-used database will be listed at the end of the DATA command GET_DATA. A user of an unsolicited database should beware when he gets such a list, and contact the database supplier.

The optional reference index {Ref. Index} is an integer number indicating where to find the particular function in a special reference file. The references are listed when doing the GET_DATA command in the DATA module. They may also be listed in the GIBBS module with the command LIST_DATA with the option R. For accounting the reference indices, see also the keyword REFERENCE_FILE.

The reference index field can also be an abbreviation (such as REF:250, REF_002, or REF-SGTE) simply denoting the original reference; in this case, the reference may not be obtained when issuing the DATA command GET_DATA or the GIBBS command LIST_DATA (with the option R).

However, since TCCN/TCW1, the references directly coded in the database definition file (**setup.TDB) that starts with a letter can be shown when issuing the DATA command GET_DATA or the GIBBS command LIST_DATA (with the option N or R). Normally, such references must be located after the LIST_OF_REFERENCE keyword. It is recommended to use reference code names such as REF001, REF018, etc. The reference list, which is generated by the GIBBS command LIST_DATA <file> with the N or R option, is thus also possible to be directly read by the DATA module.

The DATA module can selectively retrieve functions which are necessary for a defined system from a database that has functions stored in its setup file or SEQ sequential function file, while all other functions irrelevant for the defined system are simply ignored and will not be saved in associated GIBBS and POLY workspaces. Previously, this can only be done for large databases that have functions stored in RND1 random or FTP function files.

Examples:

```
FUNCTION GFREE 298.15 1000+GFUNXY#; 6000 N !
FUNCTION GFUNXY 298.15 -1000+200*T+30*T*LOG(T); 6000 N 505 !
FUNCTION G0_CAO 298.15 -663538.11+352.67749*T-57.7533*T*LN(T)
+5.3895E-03*T**2-8.879385E-07*T**3+575530*T**(-1);
1400.00 Y -625196.99+78.896993*T-20.40145*T*LN(T)
-1.112923E-02*T**2+5.1896733E-07*T**3-6917350*T**(-1);
2900.00 Y -499226.55-490.37695*T+51.95912*T*LN(T)
-2.961051E-02*T**2+1.4033905E-06*T**3-48114685*T**(-1);
3172.00 Y -587711.89+375.04117-62.76*T*LN(T);
6000.00 N REF020 !
```

3.15 PARAMETER

```
PARAMETER [GIBBS parameter name] [lowest temp. limit]
[expression 1]; [upper temp. limit 1] Y
[expression 2]; [upper temp. limit 2] Y
[expression 3]; [upper temp. limit 2] Y
..... ; ..... Y
```


[expression n-1]; [upper temp. limit n-1] Y
 [expression n]; [upper temp. limit n] N {Ref. Index} !

The PARAMETER keyword can appear in both files for database definition and sequential storage, but not in FTP files. After the keyword PARAMETER, a valid **GIBBS parameter name** should be given.

It is used for defining standard Gibbs energies (*i.e.*, the G parameters for Gibbs energy of formations) of all valid end-members of various stoichiometric and solution phases, and excess Gibbs energies (*i.e.*, the L parameters for Gibbs energy of interactions) of all binary, ternary, quaternary or higher-order interactions in various solution phases; both standard Gibbs energies and excess energies can also have parameters for contributions from PT-dependent volume variations (*i.e.*, the V0, VA, VB, VC and VK parameters for molar volume, thermal expansivity, bulk modulus, isothermal compressibility and high-pressure fitting parameter), magnetic ordering (*i.e.*, the TC and BM parameters for Curie temperature and Bohr magneton number) and hypothetical electrostatic interactions (*i.e.*, BM parameter for Born functions $\omega_{Pr,Tr}$ of aqueous solute species).

The general form of a parameter is:

<identifier>(<phase>, <constituent array>; <digit>) <xxx> <expression> <yyy>
 <keyword Y or N> <zzz> !

where

identifier	is the parameter type;
phase	is the phase name (maximum 24 characters);
constituent array	is the specific constituent array in the phase;
digit	is the degree of composition-dependent interaction contribution (an integer number from 0 through 9), that is only for excess energy (L), Curie temperature (TC) and Bohr magneton number (BMAGN), as well as for volume-related parameters (V0 or VA or VB or VC or VK); if it is valued as zero, or if it is for the standard Gibbs energy (G) for which the degree is always zero, it can be omitted;
expression	is the mathematical relation to describe the parameter;
xxx and yyy	are the low and high temperature limits respectively for the applicable temperature range of the parameter expression;
keyword Y or N	is the indicator on if there is continuation for the parameter expression or not;
zzz	is the reference index/number for the assessment of this parameter; and
sign “!”	is used to indicate that the current parameter definition is ended.

The **GIBBS parameter name** has a general form of:


<identifier>(<phase>,<constituent array>;<digit>)

Examples of parameter names:

G (GAS, C1O2)	The Gibbs energy of formation of a CO2 molecule in gas.
G (FCC, Fe:VA)	The Gibbs energy of formation of fcc Fe with interstitials.
L (LIQ, Fe, Cr; 0)	The regular solution parameter for Fe and Cr in liquid.
L (LIQ, Fe, Cr; 1)	The sub-regular solution parameter.
TC (BCC, Fe:Va)	The Curie temperature of bcc Fe.
BMAGN (BCC, Fe:Va)	The Bohr magneton number parameter of bcc Fe.

The **GIBBS parameter name** consists of several parts. The first is a *type-identifier*. The following *type-identifiers* are legal:

☞ G	Standard energy parameter (Gibbs energy of formation);
☞ L	Excess energy parameter (Gibbs energy of interaction);
☞ TC	Curie temperature for magnetic ordering;
☞ BMAGN or BM	Bohr magneton number for magnetic ordering (or Born function $\omega_{Pr,Tr}$ for aqueous solute species).
☞ V0	Molar volume at 298.15 K and 1 bar (a numeric value only);
☞ VA	Integrated thermal expansivity $\int_{298.15}^T \alpha(T) dT$;
☞ VB	Bulk modulus at 1 bar;
☞ VC	Isothermal compressibility;

 VK High-pressure fitting parameter.

One may also use G for interaction parameters; and on output list (performed by the GIBBS command LIST_PARAMETER or LIST_PHASE_DATA) the type-identifier L is always used for interaction parameters. Note that the type-identifier BM is also used for Born functions $\omega_{Pr,Tr}$ of aqueous solute species.

The identifier must be followed by an opening parenthesis, a phase name, a comma and a constituent array. Optionally, the constituent array can be followed by a semicolon and a digit. The parameter name is terminated by a closing parenthesis.

Specifying the phase name always in UPPER-cases is recommended; however, if one would prefer to write it as a mixture of UPPER-case and lower-case in a database, the DATA module will then automatically convert all lower-cases to UPPER-cases, as the GIBBS module only recognizes UPPER-case phase names. It is important that if a phase bears a legal phase-type (among G, A, Y, L, I, F and B) in its phase definition (already by the PHASE keyword; such as GAS:G, LIQUID:L, SLAG:L, IONIC_LIQ:Y, SPINEL:I, FCC_L12:F, HCP_D021:F, BCC_B2:B, AQUEOUS:A), such a valid phase-type code should not be attached to the phase name in the PARAMETER keyword.

The constituent array consists of a list of constituent names. Interaction parameters have two or more constituents from the same sublattice separated by a comma. If the phase has sublattices, at least one constituent in each sublattice must be specified. The constituents in different sublattices must be given in sublattice order and are separated by a colon.

After the component array, a sub-index digit can be specified after a semicolon. This digit must be in the range 0 to 9. The interpretation of the sub-index depends on the excess energy model used for the phase. If no semicolon and digit are given, the sub-index value is assumed to be as zero.

The excess energy parameters, e.g., the regular/subregular (binary) parameter or ternary parameters, are multiplied with two or more fractions of the constituents from the same sublattice of the solution phase. These additional constituents must be given as interacting constituents. Note that solution phases with sublattices may have interacting constituents in each sublattice.

One may use an asterisk, “*”, to denote that the excess interaction parameter is independent of the constituents of a specific sublattice. For example, L(FCC_L12,AL,NI:*) means that the interaction parameter is for the binary interaction between constituents AL and NI on the first sublattice in the FCC_L12 solution phase, while it is independent of all constituents on the second sublattice. A interaction parameter in the list of constituents is always added to the Gibbs energy and the asterisk “*” is calculated with the term of $[1-\sum Y(\text{specified constituents})]$, which implies that in an A-B binary system the following three L parameters are identical (but in higher-order systems, they are different):

L(phase,A,B)	is multiplied with $X(A) * X(B)$
L(phase,A,*)	is multiplied with $X(A) * (1-X(A))$
L(phase,B,*)	is multiplied with $X(B) * (1-X(B))$

A parameter always starts with a lowest temperature limit of its applicability, followed by one or more (up to 10) expressions (TP-Functions) that are coded as mathematical relations of constants, functions of stable variables (T and P) and entered functions (normally with a # suffix, e.g. +3*GSERAL#). However, functions no longer need to be suffixed with #. For further information, see *Section 11.2.5* (Functions of Temperature and Pressure) in the *Thermo-Calc Console Mode, Version User's Guide* document. The expression is a FORTRAN-like expression and operators +, -, *, = and ** can be used (** only with integer powers). Unary-functions LN or LOG (both for natural logarithm) and EXP (for exponential) can also be used. Each expression (TP-Function) should ends with a semicolon “;”, and be followed by its upper applicable temperature limit and a continuation indicator (Y for continuing with the next expression, or N for ending the parameter's expression). If there is no continuation after a specific expression (TP-Function), the reference index can optionally be given after the N indicator.

A complete/valid parameter entry can be written in several continuation lines if the parameter's expression (TP-Function) is too long or if there are more than one applicable expressions (TP-Function), as the maximum length of each line is 78 characters. Similarly (see *Section 3.14*), it is recommended to **always have at least one empty space at the beginning of each continuation line**. One should avoid entering parameters like

```
PARAMETR G(LIQUID,A,B) 298.15
-2000+4568*T+2*GFUNAB#; 6000 N !
```

Such a parameter will be read by the DATA module as 2000+4568*T+2*GFUNAB#, rather than as -2000+4568*T+2*GFUNAB#.

This mistake can be avoided by giving at least one empty space as the first character of a new line, such as

```
PARAMETR G(LIQUID,A,B) 298.15
-2000+4568*T+2*GFUNAB#; 6000 N !
```

The lowest-temperature limit (in Kelvin) for the applicability of the (first) TP-Function in a parameter is normally set by default as 298.15 K, in most cases; however, one may set another limit when it is applicable (according to experimental data and assessments). An upper-temperature limit (in Kelvin; followed by an Y or N sign) for the applicability of each TP-Function in a parameter must be given after the semicolon ";" sign immediately following the specific TP-Function; and the highest-temperature limit (in Kelvin) for the applicability of the current parameter is always followed by the N sign. If a negative number is given as the lowest-temperature limit, it will be assumed that there are breakpoints in pressure for this parameter; in such a case, it is interpreted as the lowest-pressure limit (in Pascal), and the other limits in the current parameter will also be taken as pressure limit values (in Pascal).

The temperature/pressure limits for the parameters are checked during calculations. An indicator will be set if the actual temperature/pressure condition is below the lowest temperature/pressure limit or above the highest temperature/pressure limit; and in such a case, an extrapolation will be conducted using the TP-Function valid in the nearest temperature/pressure range.

The optional reference index {Ref. Index} is an integer number indicating where to find the particular parameter in a special reference file. The references are listed when doing the GET_DATA command in the DATA module. They may also be listed in the GIBBS module with the command LIST_DATA with the option R or N. For accounting the reference indices, see also the keyword REFERENCE_FILE.

The reference index field can also be an abbreviation (such as REF:250, REF_002, or REF-SGTE) simply denoting the original reference; in this case, the reference may not be obtained when issuing the DATA command GET_DATA or the GIBBS command LIST_DATA (with the option R or N).

However, since TCCN/TCW1, the references directly coded in the database definition file (**setup.TDB) that starts with a letter can be shown when issuing the DATA command GET_DATA or the GIBBS command LIST_DATA (with the option R or N). Normally, such references must be located after the LIST_OF_REFERENCE keyword. It is recommended to use reference code names such as REF001, REF018, etc. The reference list, which is generated by the GIBBS command LIST_DATA <file> with the N option, is thus also possible to be directly read by the DATA module.

Examples:

```
PARAMETER G(BCC,FE:VA) 298.15 1000+200*T+...; 6000 N 91DIN !
PARAMETER TC(BCC,FE:VA) 298.15 +1043; 6000 N 91DIN !
PARAMETER BMAGN(BCC,FE:VA) 298.15 +2.22; 6000 N 91DIN !
PARAMETER G(SIGMA,FE:CR:CR;0) 298.15 1000+200*T+...; 6000 N 101 !
PARAMETER G(LIQUID,AL;0) 298.15 +11005.553-11.840873*T
+7.9401E-20*T**7+GHSERAL#;
933.60 Y +10481.974-11.252014*T+1.234264E+28*T**(-9)+GHSERAL#;
2900.00 N REF:283 !
PARAMETER G(BCC_A2,PB:C) 298.15 UN_ASS#; 300 N REF:0 !
PARAMETER G(BCC_A2,NI:C;0) 298.15 +GHSERNI#+3*GHSERCC#
+400000-100*T; 6000 N REF071 !
PARAMETER G(BCC_A2,MN:VA) 298.15 +GMNBCC#; 6000 N REF285 !
PARAMETER BM(AQUEOUS,OH-1) 298.15 +Z0002PW0#; 1600 N 155 !
PARAMETER L(BCC,FE,CO:VA;0) 298.15 1000+200*T+...; 6000 N !
PARAMETER L(BCC,FE,CO:VA;1) 298.15 1000+200*T+...; 6000 N !
PARAMETER L(BCC,FE,CO:VA;2) 298.15 1000+200*T+...; 6000 N !
PARAM TC(BCC_A2,CO,MO:VA;0) 298.15 -3700; 6000 N R454 !
PARAM TC(BCC_A2,CO,MO:VA;1) 298.15 +2300; 6000 N R454 !
PARAM BMAGN(BCC_A2,CO,MO:VA;0) 298.15 -3.445; 6000 N R454 !
PARAM V0(BCC_A2,CR,FE:VA;0) 298.15 +ZERO#; 6000 N REF06V !
PARAM V0(BCC_A2,CR,FE:VA;1) 298.15 -1.10524097E-7; 6000 N REF06V !
PARAM V0(BCC_A2,CR,FE:VA;2) 298.15 +1.40024130E-7; 6000 N REF06V !
PARAM VA(BCC_A2,CR,FE:VA;0) 298.15 -6.49444634E-6*DELTAT#; 6000 N REF06V !
PARAM VA(BCC_A2,CR,FE:VA;1) 298.15 +2.91269321E-5*DELTAT#; 6000 N REF06V !
```

3.16 OPTIONS

OPTIONS /[alloy name] ([composition limitations for all alloying elements]) !

The **OPTIONS** keyword defines an "alloy" in a database. It has been available since TCCM. An alloy has a name, a major component and a number of alloying elements. The purpose for defining an alloy is to be able to inform the user about applicable composition limits of the current database in applications to that particular type of alloy. It is possible to have several alloys in the same database. The alloys are given after the **OPTIONS** keyword in the database.

The alloy name must be preceded by a slash "/" and terminated by the opening parenthesis, and no spaces are allowed between. The alloy name is maximum 8 characters. After the parenthesis, follows the major element and a parenthesis with its minimum mass and minimum mole percent given inside. Then, follows the alloying element names, each with its maximum mass and mole percent given within parenthesis. There must a space between definitions for each alloying element. The alloy definition is terminated by a closing parenthesis, and the whole **OPTIONS** keyword by the exclamation mark "!".

Example:

```
OPTIONS /SSteel(Fe(60,60) CR(30,30) NI(15,15) SI(1,1) N(.1,1)) !
```

3.17 TABLE

TABLE [name]*8 [start temp] [end temp] [delta temp] [table values] !

The **TABLE** keyword can appear in both files for database definition and sequential storage, but not in FTP files. It makes a table of Gibbs energy as a function of temperature where the values are given from the start temperature to the end temperature, at a step of the delta temperature.

It is also recommended to *always have at least one empty space at the beginning of each continuation line*. Otherwise, the **DATA** module may issue some error messages when reading the table entry.

Example:

```
TABLE DEMENTAB 1000.0 1500.0 100.0 -2912.9008 -2834.2416 -2755.5824
-2677.7600 -2600.7744 -2524.2072 !
```

3.18 ASSESSED_SYSTEMS

ASSESSED_SYSTEM [descriptions on special treatments for specific assessed systems]*8000 !

The keyword `ASSESSED_SYSTEMS` can be included in the database definition file (the `***setup.TDB` file). A maximum of 8000 characters after the keyword (to describe some special options when the `DATA`, `GIBBS` and `POLY` models deal with the existing systems with assessed data) is allowed until the exclamation mark “!”.

Since `TCCP/TCW2` and `DICTRA22`, several `ASSESSED_SYSTEMS` keywords can be used in the same `TDB` file in order to have more lists of assessed systems; in other word, the number of the `ASSESSED_SYSTEMS` keyword entries in the same `TDB` file is not limited.

The assessed systems in the database, and their special treatment options, are typed after the keyword. The elements (should always be in `UPPER CASE`) in each assessed system must be in alphabetical order and be separated by a hyphen, like `C-FE` for the Fe-C binary system, and `C-CR-FE` for the Fe-Cr-C ternary system. A space must be between each assessed system. Information on assessed binary, ternary or higher-order systems may also be given in this way. Note that a ternary system like `C-CR-FE` does not imply that the binary `C-CR`, `C-FE` and `CR-FE` are assessed! There is no way to indicate partially assessed systems.

There is a field to give some descriptive information for a specific system, with various options on:

- ◆ How to reject or restore phase(s) from the current database in the `DATA` module;
- ◆ How to set major constituent(s) in the first composition set and to set a second composition set for a specific phase available in the current database in the `GIBBS` module; *and*
- ◆ How to calculate this specific system in the `POLY` module.

This is the facility used by the `BIN` (binary phase diagrams) and `TERN` (ternary phase diagrams) modules in the Thermo-Calc software/database package.

The descriptive information must immediately follow the specific system name, and must be enclosed within parenthesis “(” and “)”, and the left parenthesis must follow directly after the system, such as

```
AL-NI (TDB +L12 ;G5 C-S:L12/NI:AL:VA ;P3 STP:.8/1200/1 STP:.2/600/1)
```

The information is specific to Thermo-Calc, and intended to be cryptic. As in the above example,

- The syntax “`TDB`” means that the commands to the `DATA` module proceed, and “`+L12`” in the example means that the phase called `L12` should be restored (it has been rejected by default).
- The directive “`;G5`” means that the following are commands to the `GIBBS` module. “`C_S:`” means creating a second composition set, after the colon follows the phase name and after the slash the major constituents.
- After the directive “`;P3`” follows commands to the `POLY` module. “`STP:`” means setting a start point with the value of the X-axis first (composition for the second element in a binary system), the slash separates the Y-axis value (temperature), and possibly one or more directions (-1, 1, -2 or 2).

A summary of the allowed syntax is:

- `TDB` accepts
`+phase` and `-phase` for restore/reject.
- `;G5` accepts
`MAJ:phase/constituent-array` for major constituents of the first composition set.
`C_S:phase/constituent-array` for a second composition set.
- `;P3` accepts
`TMM:lt/ht` for the low-/high-temperature limits (`lt` and `ht`; for instance `TMM:500/4000`) suitable for calculating phase diagrams and property diagrams of a binary system.
`*` for a default start point which is set as
for a binary system: at the composition $X(2^{\text{nd}} \text{ element}) = .1234$, temperature $T = 1100 \text{ K}$ and with the default directions; *or*
for a ternary system: at the compositions $X(2^{\text{nd}} \text{ element}) = .1234$ and $X(3^{\text{rd}} \text{ element}) = .1234$ and with the default directions.
`STP:x/t/d1/d2/d3` for a specific start point in a binary system which is set as at the composition $X(2^{\text{nd}} \text{ element}) = x$ and temperature $T = t$ (in K), and with the directions `d1`, `d2` and/or `d3`.

STP:x1/x2/d1/d2/d3 for a specific start point in a ternary system which is set as at the compositions $X(2^{\text{nd}} \text{ element})=x1$ and $X(3^{\text{rd}} \text{ element})=x2$ and with the directions d1, d2 and/or d3.

The direction(s) can be defined as -1, 1, -2 or 2. If no direction is specified, all default directions will be used (meaning no ADD command will be enforced in the POLY module).

If only one start point is specified, the direction(s) may be omitted; if more than one start points are specified, at least one direction for each start point must be given for all start points.

Each entry for a specific binary or ternary sub-system can be written in one or several lines (each line with 78 characters).

Some other examples given below may clarify:

```

ASSESSED_SYSTEMS
AL-NI(TDB +L12 +BCC_B2 ;G5 C_S:L12/NI:AL:VA
      ;P3 STP:.8/1000/1 STP:.45/700/1 STP:.7/700/1)
AL-PB(TDB -HCP -BCC
      ;G5 MAJ:LIQ/AL MAJ:FCC/AL:VA C-S:LIQ/PB C-S:FCC/PB:VA ;P3 *)
CR-FE(;G5 C-S:BCC/CR:VA ;P3 STP:.6/1200/1/-2/2)
AG-CU(;G5 MAJ:FCC/AG:VA C_S:FCC/CU:VA ;P3 STP:.3/1000)
C-NB(;P3 STP:.9/1100/1)
C-SI(;P3 *)
CO-CR(;G5 MAJ:FCC/CO:VA C_S:FCC/CR:VA ;P3 STP:.1/1100)
CR-FE(TDB -HCP ;G5 C_S:BCC/CR:VA ;P3 STP:.6/1200/1/-2/2)
CR-NI(;P3 *)
CR-W(;G5 MAJ:BCC/W:VA C_S:BCC/CR:VA
      ;P3 TMM:500/4000 STP:.3/700/1 STP:.3/1800/2 )
CU-FE(TDB -HCP ;G5 MAJ:LIQ/CU MAJ:FCC/FE:VA C_S:FCC/CU:VA
      ;P3 STP:.9/1400)
FE-N(TDB +FE4N ;P3 *)
FE-O(TDB -LIQUID +IONIC ;G5 C_S:ION_LIQ/FE+2:O-2 MAJ:ION_LIQ/FE+2:VA
      ;P3 STP:.2/2000/1 )
FE-S(TDB -LIQUID +IONIC ;G5 C_S:ION_LIQ/FE+2:S MAJ:ION_LIQ/FE+2:S-2 ;P3 *)
AL-MG-SI(;P3 *)
C-CR-FE ;G5 MAJ:BCC/FE:VA C_S:BCC/CR:VA ;P3 *) !
    
```

Note that the semicolon “;” is a part of the “;G5” and “;P3” directives. A long descriptive information can be written in more than one line, such as for the AL-NI, AL-PB, CU-FE and FE-O systems shown above.

The directive “;P3 *” is needed if the default start point should be used. If there is no “;P3” directive, the BIN or TERN module will generate some 20 different start points in order to cover all possible compositions and temperatures (for a binary system) or all possible compositions (for a ternary system under any specific temperature).

Since TCCS/TCW5, the GIBBS-action parts (enforced by the syntaxes “MAJ:phase/constituent-array” for major constituents of the first composition set, and “C_S:phase/constituent-array” for a second composition set) and POLY-action parts (enforced by the syntax “STP:x/t/d1/d2/d3” for a specific start point in a binary system, or “STP:x1/x2/d1/d2/d3” for a specific start point in a ternary system) becomes much less important (and usually unnecessary for some cases), thanks to that the implemented Global Minimization Technique and further improved stepping/mapping routines can automatically detect and create appropriate additional composition set(s) where it is necessary and can generally handle the starting point(s) in an automatic and comprehensive way. However, if the POLY-action syntaxes “STP:x/t/d1/d2/d3” or “STP:x1/x2/d1/d2/d3” are used in the database, the Thermo-Calc software will assume that the user (database-manager) knows exactly what he is doing (regarding enforced starting points), and thus no automatic starting points will be created/used during the BIN/TERN-module calculations.

3.19 REFERENCE_FILE

REFERENCE_FILE [file name] !

The keyword `REFERENCE_FILE` takes a reference file name as its argument. This reference file (*that must be edited in a very restrictive way and must also be saved as a blocked file with a fixed line-length for each line in the file, as described below*) contains a complete list of the references for the various parameters (and sometimes functions) in the database. The file must have a fixed record structure: each reference entry with one or several records, and each record with exactly 78 characters written in one single blocked line; and no empty space is allowed at the beginning of all lines in the file. If there are more than one record entered for a reference entry, all the continuation lines must start with a “&” sign. The line number of the first record for a specific reference entry is then accounted as the unique integer for that specific reference, which will be referred when a parameter or function calls this integer as the optional {Ref. Index}. See keyword `PARAMETER` or `FUNCTION` about specifying a reference index.

Example:

```

/-1<G>                                T.C.R.A.S. Class 1
AG1.64TE1                              THERMODATA 01/93
&28/01/93
&SILVER 1.64-TELLURIDE. Solid Standard State.
AG1                                    HULTGREN SELECTED VAL.          SGTE **
&AT.WEIGHT 107.870,STANDARD STATE:CODATA KEY VALUE.MPT=1234.93K.
&--U.D. 30/10/85 .
AG1<G>                                T.C.R.A.S Class: 1
AG1/+1<G>                              T.C.R.A.S Class: 1
AG1BR1                                  N.P.L.                          SGTE **
&Tfusion uncertain and heat vaporization estimated.
AG1BR1<G>                              THERMODATA 01/93
&28/01/93
&Gaseous Standard State.
AG1BR103                               BARIN & KNACKE.SUPPL.REF:62,* SGTE **
&AGO3BR                                SILVER OXYTRIBROMIDE

```

In the above example, the unique integers of related references (for assessed elements, species, phases, interactions, *etc.*) are:

```

1  /-1<G>
2  AG1.64TE1
5  AG1
8  AG1<G>
9  AG1/+1<G>
10 AG1BR1
12 AG1BR1<G>
15 AG1BR103

```

3.20 LIST_OF_REFERENCE

LIST_OF_REFERENCE

NUMBER SOURCE

[REFxxx] '[Detailed reference]'

.....

..... !

The keyword `LIST_OF_REFERENCE` starts a reference list that is directly coded in the database definition file (`***setup.TDB`). Its argument begins on the following line, and normally has an explanation line (`NUMBER SOURCE`) that is followed by various reference codes. Each reference code may occupy one or more lines (each line with maximum 78 characters), but must have a reference code name (that starts with a letter) and the detailed reference information (that is written within two single-quotation marks, “' '”). It is recommended to use reference code names such as `REF001`, `REF018`, *etc.* A maximum of 400,000 characters after the keyword is allowed until the exclamation mark “!”.

Such a reference list can be shown when issuing the DATA command GET_DATA or the GIBBS command LIST_DATA (with the R option). The reference list, which is generated by the GIBBS command LIST_DATA <file> with the N option, has this structure, and is thus possible to be directly read by the DATA module.

Example:

```
LIST_OF_REFERENCES
NUMBER SOURCE
REF283 'Alan Dinsdale, SGTE Data for Pure Elements,
        Calphad Vol 15(1991) p 317-425,
        also in NPL Report DMA(A)195 Rev. August 1990'
REF224 'P-Y Chevalier, Thermochemica Acta, 130 (1988) p 33-41; AG-SI'
!
```

3.21 ADD_REFERENCE

ADD_REFERENCE

```
[REFxxx] '[Detailed reference]'
.....
..... !
```

With the directive LIST_OF_REFERENCES, it has been possible to include the references in the same database definition file (**setup.TDB). But for large databases the space for references has not been big enough. Therefore, the other directive called ADD_REFERENCES makes it possible to have virtually unlimited number of references. The use of this directive is the same as that of LIST_OF_REFERENCES, and must be used after the LIST_OF_REFERENCE section. Several ADD_REFERENCES section can be used in a single database, if there are too many references.

However, it is recommended to *always* start with a so-called “empty reference” (such as the DUMP0 reference in the following example) as the first entry in each of the ADD_REFERENCE sections to make sure that all references will be listed out appropriately as retrieving data by issuing the GET_DATA command.

Example:

```
ADD_REFERENCES
DUMP0 'Empty reference 0'
REF4 'J-O Andersson, B. Sundman, CALPHAD Vol 11, (1987), p 83-92
      TRITA 0270 (1986); CR-FE'
REF5 'J-O Andersson, Met. Trans A, Vol 19A, (1988) p 627-636
      TRITA 0207 (1986); C-CR-FE'
REFS 'Pingfang Shi (2006), TCS PTERN Public Ternary Alloys Database,
      v1.2; Modified L0(BCC,Fe,C) and L0(BCC,Cr,C) parameters
      at high temperatures.'
... ..
!
ADD_REFERENCES
DUMP1 'Empty reference 1'
REF275 'A. Fernandez Guillermet, Z. Metallkde. Vol 79(1988) p.524-536,
        TRITA-MAC 362 (1988); C-CO-NI AND C-CO-FE-NI'
REF393 'K. Frisk, Metall. Trans. Vol 21A (1990) p 2477-2488,
        TRITA 0409 (1989); CR-FE-N'
REF1096 'P. Gustafson, Metall. Trans. 19A(1988) p 2547-2554,
         TRITA-MAC 348, (1987); C-CR-FE-W'
... ..
!
```

3.22 CASE and ENDCASE

```
CASE [ELEMENT/SPECIE/PHASE] !
  IF (boolean algebra on element, species or phase names) THEN
    [GIBBS/POLY/DATA command] !
ENDCASE !
```

The keyword CASE takes, as its argument, a definition on which type of the following boolean algebra will operate. A simple boolean algebra using AND and OR with a maximum of 4 levels of parentheses will work. The CASE construction must end with the ENDCASE keyword. This makes it possible to have additional GIBBS or POLY commands executed depending on the user selection of elements, species or phases. The DATA commands that can be given as secondary keyword to DEFAULT_COMMAND can also be executed (see section 3.10).

Example 1:

```
CASE ELEMENT !
  IF((CR OR TI OR V) AND N)
  THEN GES AM_PH_DES @ C_S ,, CR MO TI V:C N: !
ENDCASE !
```

Example 2:

```
CASE ELEMENT !
  IF(O) THEN TDB DEFINE_SYSTEM_ELEMENT /- !
ENDCASE !
```

Example 3:

```
CASE ELEMENT !
  IF(AL AND FE)
  THEN TDB DEF_SYS_ELEMENT VA !
ENDCASE !
```

3.23 VERSION_DATA

```
VERSION_DATE [string]*78 !
```

The string is denoted as the version/revision date and database manager of the database. Nowadays, this directive is used mainly for the purpose of keeping the development and revision history mostly updated.

Example:

```
VERSION_DATE Last update and adjustment: Database Manager, 2012-06-25 !
```

4 DICTRA Extensions to Database Definition File Syntax

Software packages for simulation of diffusional phase transformations, such as DICTRA, need both thermodynamic data and kinetic data (*i.e.*, diffusivities or mobilities). Naturally, the handling and storage of kinetic data will also benefit from the use of some kind of database management. Thus, the TDB database definition file syntax has been extended to incorporate some new keywords needed for storing kinetic data.

4.1 PARAMETER

```
PARAMETER [special GIBBS parameter name] [low temp. limit]
  [expression 1]; [upper temp. limit 1] Y
  [expression 2]; [upper temp. limit 2] Y
  [expression 3]; [upper temp. limit 2] Y
  ..... ; ..... Y
  [expression n-1]; [upper temp. limit n-1] Y
  [expression n]; [upper temp. limit n] N {Ref. Index} !
```

The keyword `PARAMETER` allows to enter all types of normal GIBBS parameters for thermodynamic data (as previously presented in *Section 3.15*), as well as five special extensions suitable for kinetic data used in the DICTRA software. Valid extensions to special GIBBS parameter names are:

- MQ ⇒ Activation enthalpy for mobility equation.
- MF ⇒ Pre-exponential factor for mobility equation.
- DQ ⇒ Activation enthalpy for diffusivity equation.
- DF ⇒ Pre-exponential factor for diffusivity equation.
- VS ⇒ Volume counted per mole of volume carrying species.

Examples:

```
PARAMETER MQ(BCC,FE:VA) 298.15 1000+200*T+...; 6000 N !
PARAMETER MF(BCC,CO:VA) 298.15 1000+200*T+...; 6000 N !
PARAMETER DQ(FCC,FE:VA) 298.15 1043+...; 6000 N 10 !
PARAMETER DF(FCC:CR:C ) 298.15 1000+200*T+...; 6000 N 10 !
PARAMETER VS(FCC) 298.15 1000+200*T+...; 6000 N 11 !
```

4.2 DIFFUSION

```
DIFFUSION [model keyword] [phase name] [additional parameter(s)] !
```

The keyword `DIFFUSION` specifies what type of diffusion model to use for a phase if the default model is not desired. The default model calculates the full diffusion matrix. A diffusivity is calculated from the different mobilities and the thermodynamic factors. The former ones are calculated as:

$$M = \exp(\sum MF/RT) \exp(\sum MQ/RT) / RT$$

where \sum stands for a weighted summation of the different MF's and MQ's plus possibly a Redlich-Kister term. Valid model keywords are:

- NONE
no diffusion in this phase
- DILUTE
constitution list of dependent species in each sublattice must be given as an additional parameter. Only the diagonal terms in the diffusion matrix are calculated. $D = \exp(\sum DF/RT) \exp(\sum DQ/RT)$
- SIMPLE
constitution list of dependent species in each sublattice must be given as additional parameter. Only the diagonal terms in the diffusion matrix are calculated. $D = \sum DF + \sum DQ$

MAGNETIC

the so-called ALPHA and ALPHA2 parameters must be given as additional parameters. ALPHA is for the substitutional magnetic model and ALPHA2 for the interstitial one. By appending an “&” sign and a species name after the alpha keyword one can supply individual values for the different species. The full diffusion matrix is calculated.

Examples:

```
DIFFUSION NONE SIGMA !
DIFFUSION DILUTE CEMENTITE : FE : C : !
DIFFUSION MAGNETIC BCC_A2 ALPHA=0.3 ALPHA2&C=1.8 ALPHA2&N=0.6 !
```

4.3 ZERO_VOLUME_SPECIES

ZERO_VOLUME_SPECIES [list of species] !

In the DICTRA software, the assumption that the volume is carried by the substitutional elements only is applied. The interstitial elements are assumed to have zero molar volumes. The keyword ZERO_VOLUME_SPECIES uses a list of which species are to be considered as zero volume ones for an argument.

Example:

```
ZERO_VOLUME_SPECIES VA C N !
```

5 Examples of Database Definition Files

5.1 Example 1: A small steel database

```

TEMP-LIM 500.0 2000.0 !
$
$ELEMENT, NAME, REF.STATE, ATOMIC-MASS, H0, S0 !
ELEMENT VA VACUUM          0.0    0.0 0.0 !
ELEMENT C GRAPHITE        12.011 0.0 0.0 !
ELEMENT V BCC             50.9415 0.0 0.0 !
ELEMENT CR BCC-PARAMAGNETIC 51.996 0.0 0.0 !
ELEMENT FE FCC-PARAMAGNETIC 55.847 0.0 0.0 !
ELEMENT NI FCC-PARAMAGNETIC 58.69  0.0 0.0 !
ELEMENT MO BCC            95.94   0.0 0.0 !
ELEMENT W BCC             183.85  0.0 0.0 !
$
$PHASE, NAME, TYPE, NR-OF-SUBL, SITES-IN-EACH-SUBL. !
PHASE BCC      B1M  2 1.0 3.0 !
PHASE FCC      F2M  2 1.0 1.0 !
PHASE HCP      0    2 2.0 1.0 !
PHASE LIQUID   3    2 1.0 1.0 !
PHASE CEMENTITE 4    2 3.0 1.0 !
PHASE M23C6    4    2 23.0 6.0 !
PHASE M7C3     4    2 7.0 3.0 !
PHASE M6C      4    4 2.0 2.0 2.0 1.0 !
PHASE SIGMA    0    3 10.0 4.0 16.0 !
PHASE MU-PHASE 0    3 7.0 2.0 4.0 !
PHASE R-PHASE  0    3 27.0 14.0 12.0 !
PHASE GRAPHITE 4    1 1.0 !
$
$CONSTITUENT, PHASE-NAME : CONSTITUENTS !
CONSTITUENT BCC :V CR FE NI MO W:VA C: !
CONSTITUENT FCC :V CR FE NI MO W:VA C: !
CONSTITUENT HCP :CR FE NI:VA C N: !
CONSTITUENT LIQUID :C V CR FE NI MO W VA:VA C: !
CONSTITUENT CEMENTITE :CR FE:C: !
CONSTITUENT M23C6 :CR FE:C: !
CONSTITUENT M7C3 :CR FE:C: !
CONSTITUENT M6C :FE:W:FE W:C: !
CONSTITUENT SIGMA :FE:V CR MO:FE V CR MO: !
CONSTITUENT MU-PHASE :FE:MO W:FE MO W: !
CONSTITUENT R-PHASE :FE:MO :FE MO: !
CONSTITUENT GRAPHITE :C: !
$
$TYPE_DEFINITIONS:
TYPE-DEFINITION 0 SEQ TCPATH\DATA\METDATA\TC-THEREST.TDB !
TYPE-DEFINITION 1 SEQ TCPATH\DATA\METDATA\TC-BCC.TDB !
TYPE-DEFINITION 2 SEQ TCPATH\DATA\METDATA\TC-FCC.TDB !
TYPE-DEFINITION 3 SEQ TCPATH\DATA\METDATA\TC-LIQUID.TDB !
TYPE-DEFINITION 4 SEQ TCPATH\DATA\METDATA\TC-CARBIDES.TDB !
TYPE-DEFINITION M SEQ TCPATH\DATA\METDATA\TC-CURIE-BOHR.TDB !
TYPE-DEFINITION B GES AM-PH BCC MAGNETIC -1 .4 !
TYPE-DEFINITION F GES AM-PH FCC MAGNETIC -3 .28 !
$
$DEFAULT_COMMANDS:
DEFAULT-COMMAND DEF_ELEMENT VA !
DEFAULT-COMMAND REJ_SYS-CONST LIQUID 1 VA !
$DATABASE_INFORMATION:
DATABASE-INFO The following binary and ternary systems are available: '
FE-CR-NI by Hertzman'
FE-MO      Fernandez'
FE-CR-C    Andersson'
FE-W-C     Gustafson'
FE-W       Andersson & Gustafson' !

```

5.2 Example 2: A personal database for the Sb-Sn binary system

```

$
$
$ELEMENT, NAME, REF.STATE, ATOMIC-MASS, H0, S0 !
ELEM VA VACUUM          0.0  0.0 0.0 !
ELEM MG HCP(A3)        24.305 0.0 0.0 !
ELEM SB RHOMBOHEDRAL(A7) 121.75 0.0 0.0 !
ELEM SN BCT(A5)        118.69 0.0 0.0 !
ELEM /- ELECTRON-GAS    0      0   0 !
$
$SPECIES, NAME, STOICHIOMETRIC-FORMULA !
SPECIE MG1 MG1!
SPECIE MG2 MG2!
SPECIE MG2+ MG/+2!
SPECIE SB1 SB1!
SPECIE SB2 SB2!
SPECIE SB4 SB4!
SPECIE SB3- SB/-3!
SPECIE SB5- SB/-5!
SPECIE SN1 SN1!
SPECIE SN4- SN/-4!
$
$PHASE, NAME, TYPE, NR-OF-SUBL, SITES-IN-EACH-SUBL. !
PHASE BCT              Z 1 1.0!
PHASE HCP              Z 1 1.0!
PHASE RHOMBO          Z 1 1.0!
PHASE GAS:G           Z 1 1.0!
PHASE LIQUID:L        Z 1 1.0!
PHASE IONICLIQ:Y      Z 2 1 1!
PHASE SPLIQ:Y         Z 2 1 1!
PHASE BMG3SB2:I       Z 2 3 2!
PHASE AMG3SB2:I       Z 2 3 2!
PHASE MG2SN:I         Z 2 2 1!
PHASE SBSN            Z 2 1 1!
PHASE SB2SN3          Z 2 2 3!
$
$CONSTITUENT, PHASE-NAME : CONSTITUENTS !
CONSTITUENT RHOMBO :SB SN:!
CONSTITUENT HCP :MG SN:!
CONSTITUENT BCT :SB SN:!
CONSTITUENT GAS:G :MG1 MG2 SB1 SB2 SB4 SN1:
  > Gas phase, using the Ideal EOS and Mixing Model. !
CONSTITUENT LIQUID:L :SB SN:!
CONSTITUENT IONICLIQ:Y :MG2+:SB SB3- SN SN4- VA:
  > This is the Ionic Liquid Solution Phase. !
CONSTITUENT SPLIQ:Y :MG2+:SB SB3- SN SN4- VA:!
CONSTITUENT BMG3SB2:I :MG2+:SB3- SB5- VA SN4-:!!
CONSTITUENT AMG3SB2:I :MG2+:SB3- VA SN4-:!!
CONSTITUENT MG2SN:I :MG2+ VA:SB3- SN4-:!!
CONSTITUENT SBSN :SB SN:SB SN:!
CONSTITUENT SB2SN3 :SB:SN:!
$
$DEFAULT_COMMANDS:
DEFAULT-COM DEF-ELEM VA /-!
DEFAULT-COM REJ-PHASE LIQUID!
DEFAULT-COM REJ-PHASE SPLIQ!
$
$TYPE_DEFINITIONS:
TYPE-DEFINITION Z SEQ * !
$
$DATABASE_INFORMATION:
DATABASE_INFO The Sb-Sn system with isentropic temperatures!
$
$VERSION_DATE:
VERSION_DATE Last update 1986-05-18 11:39:49 !

```

Thermo-Calc Database Manager's Guide

```
$
$
$ HERE COMES THE THERMODYNAMIC DATA (expressed in functions & parameters):
$
FUNCTION MGLIQUID 298.15 -4630.90976+192.994374*T-34.0888057*T*LOG(T)
-36544605.6*T**(-2); 6000 N!
$
FUNCTION MGSOLID 298.15 -8367.34+143.677876*T-26.1849785*T*LOG(T)
+4.858E-4*T**2-1.393669E-6*T**3+78950*T**(-1);
923.00 Y -13804.4772 +202.909445*T-34.0888057*T*LOG(T)
-3.65446056E7*T**(-2) +1.06753982E28*T**(-9); 6000 N!
$
FUNCTION SBLIQUID 298.15 9071.98+146.800*T-31.38*T*LOG(T)
-2.441646E8*T**(-2); 6000 N!
$
.....
..... <more>
$
FUNCTION LFCT 298.15 -17325.6+5.03600*T; 6000 N!
FUNCTION GFCTSBSN 298.15 LFCT+SBSOLID+SNSOLID+2948.291+3721.286;
6000 N!
FUNCTION ISB 298.15 15000; 6000 N!
FUNCTION ISN 298.15 47199.9-95.6270*T; 6000 N!
$
.....
..... <more>
$
PARAMETER G(RHOMBO,SB;0) 298.15 SBSOLID; 6000 N!
PARAMETER G(RHOMBO,SN;0) 298.15 2035+SNSOLID; 6000 N!
$
PARAMETER G(HCP,MG;0) 298.15 MGSOLID; 6000 N!
PARAMETER G(HCP,SN;0) 298.15 32000+SNSOLID; 6000 N!
PARAMETER G(HCP,MG,SN;0) 298.15 -69566-9.23183*T; 6000 N!
$
PARAMETER G(BCT,SN;0) 298.15 SNSOLID; 6000 N!
PARAMETER G(BCT,SB;0) 298.15 1000+SBSOLID; 6000 N!
PARAMETER G(BCT,SB,SN;0) 298.15 0.5*ISB+0.5*ISN; 6000 N!
PARAMETER G(BCT,SB,SN;1) 298.15 0.5*ISB-0.5*ISN; 6000 N!
$
PARAMETER G(IONICLIQ,MG2+:SB3-;0) 298.15 -204389-4.98506*T
-2.75637E9*T**(-2)+3*MGLIQUID+2*SBLIQUID; 6000 N!
PARAMETER G(IONICLIQ,MG2+:SN4-;0) 298.15 -98639.5+881.073*T
-174.523*T*LOG(T)-1.79808E9*T**(-2); 6000 N!
PARAMETER G(IONICLIQ,MG2+:SB;0) 298.15 SBLIQUID; 6000 N!
$
.....
..... <more>
$
```


5.3 Example 3: A public database for the Fe-Cr-C ternary system

```

$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$ DFeCrC - TCS Demo Fe-Cr-C Alloy Solutions Database (Version 1.2)
$           only for demonstration purposes
$           used within TCC-Demo & TCW-Demo
$           (based on PDEMO/PTERN)
$ *****
$
$ Copyright @ 2004-2012: Thermo-Calc Software AB, Sweden
$
$ Edited by: Dr. Pingfang Shi, Oct. 2004 / Oct. 2006
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$ Revision history:
$   Created as PDEMO by Pingfang Shi on 2004-10-05
$   Renamed to DFeCrC by Pingfang Shi on 2006-10-25
$
$ FURTHER MODIFICATIONS:
$ =====
$ pfs: /20041005 (PDEMO)
$   * Retrieved all definitions from PTERN for the Fe-Cr-C ternary !
$
$ pfs: /20061025 (DFeCrC)
$   * Ignore ELEMENT /- definition.
$   * Modify the L(BCC_A2,FE:C,VA;0) [and L(BCC_A2,CR:C,VA;0)]
$     parameters, in order to avoid BCC-appearance (Fe-C)
$     at temperatures higher than 3900 K.
$     $L(BCC_A2,CR:C,VA;0) 298.15 -190*T; 6000 N REF1 !
$     L(BCC_A2,CR:C,VA;0) 298.15 -190*T; 3000 Y 0; 6000 N REF1 !
$     $L(BCC_A2,FE:C,VA;0) 298.15 -190*T; 6000 N REF3 !
$     L(BCC_A2,FE:C,VA;0) 298.15 -190*T; 3900 Y 0; 6000 N REF3 !
$   * Add Cr into HCP_A3 phase, and
$     add all necessary G0/L parameters (according to SSOL2/TCFE3):
$     G(HCP_A3,CR:VA;0)
$     TC(HCP_A3,CR:VA;0)
$     BMAGN(HCP_A3,CR:VA;0)
$     G(HCP_A3,FE:C;0)
$     G(HCP_A3,CR:C;0)
$     L(HCP_A3,CR:C,VA;0)
$     L(HCP_A3,CR,FE:VA;0)
$   * Add "TMM:300/3000" limits to all three binary joins in the
$     ASSESSED_SYSTEM section.
$   * Change the reference indices, e.g., 101 to REF1, in order to
$     always get the reference list correctly.
$   * Re-arrange database format slightly.
$   * Updated the DATABASE_INFO section.
$-----
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$ DATABASE INFORMATION:
$*****
$
$ DATABASE_INFO DFeCrC '
$               TCS Demo Fe-Cr-C Alloy Solutions Database'
$               (only for demonstration purposes in TCC-Demo/TCW-Demo)'
$               ***** '
$               (Version 1.2, Oct. 2006) ''
$               Copyright @ 2004-2012: Thermo-Calc Software AB, Sweden ''
$ The DFeCrC database (previously called PDEMO) is the TCS Demo Fe-Cr-C '
$ Alloy Solutions Database, which is especially designed for primarily '
$ demonstrating the uses of: '
$   * TCC-Demo software: BIN, TERN, SCHEIL and some other modules; '

```


Thermo-Calc Database Manager's Guide

```

$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$ TYPE_DEFINITIONS for phase descriptions:
$-----
$... For magnetic contributions:
  TYPE_DEFINITION A GES AMEND_PHASE_DES @ MAGNETIC -1.0 4.00000E-01 !
  TYPE_DEFINITION B GES AMEND_PHASE_DES @ MAGNETIC -3.0 2.80000E-01 !

$... For Cr/Fe-C in FCC phase and Cr/Fe-C in HCP phase:
  TYPE_DEFINITION C IF(C) THEN GES AMEND_PHASE_DES @ C_S ,, :C: !

$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$ Default Commands:
$-----
$DEFINE_SYSTEM_DEFAULT SPECIES 2 !
  DEFINE_SYSTEM_DEFAULT ELEMENT 2 !

  DEFAULT_COMMAND DEF_SYS_ELEMENT VA !

$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$ FUNCTIONS for various phases:
$-----
$
FUNCT GHSERCC 298.15 -17368.441+170.73*T-24.3*T*LN(T)
  -4.723E-04*T**2+2562600*T**(-1)-2.643E+08*T**(-2)
  +1.2E+10*T**(-3); 6000 N!
FUNCT GHSERCR 298.15 -8856.94+157.48*T-26.908*T*LN(T)
  +.00189435*T**2-1.47721E-06*T**3+139250*T**(-1); 2180 Y
  -34869.344+344.18*T-50*T*LN(T)-2.88526E+32*T**(-9); 6000 N!
FUNCT GHSERFE 298.15 +1225.7+124.134*T-23.5143*T*LN(T)
  -.00439752*T**2-5.8927E-08*T**3+77359*T**(-1); 1811.00 Y
  -25383.581+299.31255*T-46*T*LN(T)+2.29603E+31*T**(-9); 6000 N!

FUNCT GFELIQ 298.15 +GHSERFE#+12040.17-6.55843*T
  -3.6751551E-21*T**7; 1811 Y
  -10839.7+291.302*T-46*T*LN(T); 6000 N!
FUNCT GFEFCC 298.15 +GHSERFE#-1462.4+8.282*T
  -1.15*T*LN(T)+6.4E-04*T**2; 1811 Y
  -27098.266+300.25256*T-46*T*LN(T)+2.78854E+31*T**(-9); 6000 N!
FUNCT GFECM 298.15 +GPCM1#-10745+706.04*T-120.6*T*LN(T); 6000 N!
FUNCT GFEM23C6 298.15 +7.666667*GFECM#-1.666667*GHSERCR#
  +66920-40*T; 6000 N!

FUNCT GCRFCC 298.15 +GHSERCR#+7284+.163*T; 6000 N!
FUNCT GCRM23C6 298.15 -521983+3622.24*T-620.965*T*LN(T)
  -.126431*T**2; 6000 N!
FUNCT GCRM3C2 298.15 -100823.8+530.66989*T-89.6694*T*LN(T)
  -.0301188*T**2; 6000 N!
FUNCT GCRM7C3 298.15 -201690+1103.128*T-190.177*T*LN(T)
  -.0578207*T**2; 6000 N!

FUNCT GPCLIQ 298.15 +YCLIQ#*EXP(ZCLIQ#); 6000 N!
FUNCT ACLIQ 298.15 +2.32E-05*T+2.85E-09*T**2; 6000 N!
FUNCT BCLIQ 298.15 +1+3.2E-10*P; 6000 N!
FUNCT CCLIQ 298.15 1.6E-10; 6000 N!
FUNCT DCLIQ 298.15 +1*LN(BCLIQ#); 6000 N!
FUNCT ECLIQ 298.15 +1*LN(CCLIQ#); 6000 N!
FUNCT VCLIQ 298.15 +7.626E-06*EXP(ACLIQ#); 6000 N!
FUNCT XCLIQ 298.15 +1*EXP(.5*DCLIQ#)-1; 6000 N!
FUNCT YCLIQ 298.15 +VCLIQ#*EXP(-ECLIQ#); 6000 N!
FUNCT ZCLIQ 298.15 +1*LN(XCLIQ#); 6000 N!

FUNCT GPCGRA 298.15 +YCGRA#*EXP(ZCGRA#); 6000 N!
FUNCT ACGRA 298.15 +2.32E-05*T+2.85E-09*T**2; 6000 N!

```

Thermo-Calc Database Manager's Guide

```

FUNCT BCGRA      298.15 +1+3.6E-10*P;          6000 N!
FUNCT CCGRA      298.15 3.3E-10;              6000 N!
FUNCT DCGRA      298.15 +1*LN (BCGRA#);        6000 N!
FUNCT ECGRA      298.15 +1*LN (CCGRA#);        6000 N!
FUNCT VCGRA      298.15 +5.259E-06*EXP (ACGRA#); 6000 N!
FUNCT XCGRA      298.15 +1*EXP (.9166667*DCGRA#)-1; 6000 N!
FUNCT YCGRA      298.15 +VCGRA#*EXP (-ECGRA#); 6000 N!
FUNCT ZCGRA      298.15 +1*LN (XCGRA#);        6000 N!

FUNCT GPCFCC     298.15 +YCFCC#*EXP (ZFEFCC#); 6000 N!
FUNCT ACFCC      298.15 +1.44E-04*T;          6000 N!
FUNCT VCFCC      298.15 +1.031E-05*EXP (ACFCC#); 6000 N!
FUNCT YCFCC      298.15 +VCFCC#*EXP (-EFEFCC#); 6000 N!

FUNCT GPCRLIQ    298.15 +YCRLIQ#*EXP (ZCRLIQ#); 6000 N!
FUNCT ACRLIQ     298.15 +1.7E-05*T+9.2E-09*T**2; 6000 N!
FUNCT BCRLIQ     298.15 +1+4.65E-11*P;        6000 N!
FUNCT CCRLIQ     298.15 3.72E-11;            6000 N!
FUNCT DCRLIQ     298.15 +1*LN (BCRLIQ#);      6000 N!
FUNCT ECRLIQ     298.15 +1*LN (CCRLIQ#);      6000 N!
FUNCT VCRLIQ     298.15 +7.653E-06*EXP (ACRLIQ#); 6000 N!
FUNCT XCRLIQ     298.15 +1*EXP (.8*DCRLIQ#)-1; 6000 N!
FUNCT YCRLIQ     298.15 +VCRLIQ#*EXP (-ECRLIQ#); 6000 N!
FUNCT ZCRLIQ     298.15 +1*LN (XCRLIQ#);      6000 N!

FUNCT GPCRBCC    298.15 +YCRBCC#*EXP (ZCRBCC#); 6000 N!
FUNCT ACRBCC     298.15 +1.7E-05*T+9.2E-09*T**2; 6000 N!
FUNCT BCRBCC     298.15 +1+2.6E-11*P;        6000 N!
FUNCT CCRBCC     298.15 2.08E-11;            6000 N!
FUNCT DCRBCC     298.15 +1*LN (BCRBCC#);      6000 N!
FUNCT ECRBCC     298.15 +1*LN (CCRBCC#);      6000 N!
FUNCT VCRBCC     298.15 +7.188E-06*EXP (ACRBCC#); 6000 N!
FUNCT XCRBCC     298.15 +1*EXP (.8*DCRBCC#)-1; 6000 N!
FUNCT YCRBCC     298.15 +VCRBCC#*EXP (-ECRBCC#); 6000 N!
FUNCT ZCRBCC     298.15 +1*LN (XCRBCC#);      6000 N!

FUNCT GPFELIQ    298.15 +YFELIQ#*EXP (ZFELIQ#); 6000 N!
FUNCT AFELIQ     298.15 +1.135E-04*T;          6000 N!
FUNCT BFELIQ     298.15 +1+4.98009787E-12*P+3.20078924E-14*T*P; 6000 N!
FUNCT CFELIQ     298.15 +4.22534787E-12+2.71569924E-14*T; 6000 N!
FUNCT DFELIQ     298.15 +1*LN (BFELIQ#);      6000 N!
FUNCT EFELIQ     298.15 +1*LN (CFELIQ#);      6000 N!
FUNCT VFELIQ     298.15 +6.46677E-06*EXP (AFELIQ#); 6000 N!
FUNCT XFELIQ     298.15 +1*EXP (.8484467*DFELIQ#)-1; 6000 N!
FUNCT YFELIQ     298.15 +VFELIQ#*EXP (-EFELIQ#); 6000 N!
FUNCT ZFELIQ     298.15 +1*LN (XFELIQ#);      6000 N!

FUNCT GPFEFCC    298.15 +YFEFCC#*EXP (ZFEFCC#); 6000 N!
FUNCT AFEFCC     298.15 +7.3097E-05*T;          6000 N!
FUNCT BFEFCC     298.15 +1+3.25236341E-11*P+3.36607808E-16*T*P; 6000 N!
FUNCT CFEFCC     298.15 +2.62285341E-11+2.71455808E-16*T; 6000 N!
FUNCT DFEFCC     298.15 +1*LN (BFEFCC#);      6000 N!
FUNCT EFEFCC     298.15 +1*LN (CFEFCC#);      6000 N!
FUNCT VFEFCC     298.15 +6.688726E-06*EXP (AFEFCC#); 6000 N!
FUNCT XFEFCC     298.15 +1*EXP (.8064454*DFEFCC#)-1; 6000 N!
FUNCT YFEFCC     298.15 +VFEFCC#*EXP (-EFEFCC#); 6000 N!
FUNCT ZFEFCC     298.15 +1*LN (XFEFCC#);      6000 N!

FUNCT GPFEBCC    298.15 +YFEBCC#*EXP (ZFEBCC#); 6000 N!
FUNCT AFEBCC     298.15 +2.3987E-05*T+1.2845E-08*T**2; 6000 N!
FUNCT BFEBCC     298.15 +1+2.80599565E-11*P+3.06481523E-16*T*P; 6000 N!
FUNCT CFEBCC     298.15 +2.20949565E-11+2.41329523E-16*T; 6000 N!
FUNCT DFEBCC     298.15 +1*LN (BFEBCC#);      6000 N!
FUNCT EFEBCC     298.15 +1*LN (CFEBCC#);      6000 N!
FUNCT VFEBCC     298.15 +7.042095E-06*EXP (AFEBCC#); 6000 N!
FUNCT XFEBCC     298.15 +1*EXP (.7874195*DFEBCC#)-1; 6000 N!

```


Thermo-Calc Database Manager's Guide

```

PARAM BMAGN(FCC_A1,FE:VA;0) 298.15 -2.1; 6000 N REF0 !

PARAM G(FCC_A1,CR:C;0) 298.15 +GHSECR#+GHSECC#
                        +1200-1.94*T; 6000 N REF2 !
PARAM G(FCC_A1,FE:C;0) 298.15 +GFEFCC#+GHSECC#+GPCFCC#
                        +77207-15.877*T; 6000 N REF3 !
PARAM TC(FCC_A1,FE:C;0) 298.15 -201; 6000 N REF3 !
PARAM BMAGN(FCC_A1,FE:C;0) 298.15 -2.1; 6000 N REF3 !

PARAM L(FCC_A1,CR:C,VA;0) 298.15 -11977+6.8194*T; 6000 N REF2 !
PARAM L(FCC_A1,FE:C,VA;0) 298.15 -34671; 6000 N REF3 !
PARAM L(FCC_A1,CR,FE:VA;0) 298.15 +10833-7.477*T; 6000 N REF4 !
PARAM L(FCC_A1,CR,FE:VA;1) 298.15 +1410; 6000 N REF4 !
PARAM L(FCC_A1,CR,FE:C;0) 298.15 -74319+3.2353*T; 6000 N REF2 !

$***** BCC_A2 (Solution) *****
PHASE BCC_A2 %A 2 1 3 !
CONST BCC_A2 : CR%,FE%
              : C,VA% : !

PARAM G(BCC_A2,CR:VA;0) 298.15 +GHSECR#+GPCRBCC#; 6000 N REF0 !
PARAM TC(BCC_A2,CR:VA;0) 298.15 -311.5; 6000 N REF0 !
PARAM BMAGN(BCC_A2,CR:VA;0) 298.15 -.01; 6000 N REF0 !
PARAM G(BCC_A2,FE:VA;0) 298.15 +GHSEFE#+GPFEBCC#; 6000 N REF0 !
PARAM TC(BCC_A2,FE:VA;0) 298.15 1043; 6000 N REF0 !
PARAM BMAGN(BCC_A2,FE:VA;0) 298.15 2.22; 6000 N REF0 !

PARAM G(BCC_A2,CR:C;0) 298.15 +GHSECR#+3*GHSECC#+GPCRBCC#+3*GPCGRA#
                        +416000; 6000 N REF1 !
PARAM TC(BCC_A2,CR:C;0) 298.15 -311.5; 6000 N REF1 !
PARAM BMAGN(BCC_A2,CR:C;0) 298.15 -.008; 6000 N REF1 !
PARAM G(BCC_A2,FE:C;0) 298.15 +GHSEFE#+3*GHSECC#+GPFEBCC#+3*GPCGRA#
                        +322050+75.667*T; 6000 N REF3 !
PARAM TC(BCC_A2,FE:C;0) 298.15 1043; 6000 N REF3 !
PARAM BMAGN(BCC_A2,FE:C;0) 298.15 2.22; 6000 N REF3 !

$ PF-20061025: Modify the L(BCC_A2,FE:C,VA;0) [and L(BCC_A2,CR:C,VA;0)]
$ parametera, in order to avoid BCC-appearance (Fe-C)
$ at temperatures higher than 3900 K.
$PARAM L(BCC_A2,CR:C,VA;0) 298.15 -190*T; 6000 N REF1 !
PARAM L(BCC_A2,CR:C,VA;0) 298.15 -190*T; 3000 Y 0; 6000 N REFS !
$PARAM L(BCC_A2,FE:C,VA;0) 298.15 -190*T; 6000 N REF3 !
PARAM L(BCC_A2,FE:C,VA;0) 298.15 -190*T; 3900 Y 0; 6000 N REFS !
$
PARAM L(BCC_A2,CR,FE:VA;0) 298.15 +20500-9.68*T; 6000 N REF4 !
PARAM TC(BCC_A2,CR,FE:VA;0) 298.15 1650; 6000 N REF4 !
PARAM TC(BCC_A2,CR,FE:VA;1) 298.15 550; 6000 N REF4 !
PARAM BMAGN(BCC_A2,CR,FE:VA;0) 298.15 -.85; 6000 N REF4 !

PARAM L(BCC_A2,CR,FE:C;0) 298.15 -1250000+667.7*T; 6000 N REF2 !
PARAM TC(BCC_A2,CR,FE:C;0) 298.15 1650; 6000 N REF5 !
PARAM TC(BCC_A2,CR,FE:C;1) 298.15 550; 6000 N REF5 !
PARAM BMAGN(BCC_A2,CR,FE:C;0) 298.15 -.85; 6000 N REF5 !

$***** HCP_A3 (Solution) *****
$PHASE HCP_A3 %BC 2 1 .5 !
$ Note the C TYPE_DEF for 2nd HCP composition-set (M2C) is not necessary
$ for the (Fe,Cr)-C system.
PHASE HCP_A3 %B 2 1 .5 !
CONST HCP_A3 : CR,FE,
              : VA%,C : !

PARAM G(HCP_A3,CR:VA;0) 298.15 +GHSECR#+4438; 6000 N REF1 !
PARAM TC(HCP_A3,CR:VA;0) 298.15 -1109; 6000 N REF1 !
PARAM BMAGN(HCP_A3,CR:VA;0) 298.15 -2.46; 6000 N REF1 !
PARAM G(HCP_A3,FE:VA;0) 298.15 +GHSEFE#+GPFEHCP#

```

Thermo-Calc Database Manager's Guide

```

-3705.78+12.591*T-1.15*T*LN(T)+6.4E-04*T**2; 1811 Y
      +GHSERFE# +GPFHEHCP#
-3957.199+5.24951*T+4.9251E+30*T**(-9);          6000 N REF0 !
PARAM G(HCP_A3,CR:C;0) 298.15 +GHSERCR#+.5*GHSECC#
      -18504+9.4173*T-2.4997*T*LN(T)+.001386*T**2; 6000 N REF1 !
PARAM G(HCP_A3,FE:C;0) 298.15 +GFEFCC#+.5*GHSECC#+GPCFCC#
      +52905-11.9075*T;          6000 N REF3 !

PARAM L(HCP_A3,CR:C,VA;0) 298.15 +4165; 6000 N REF1 !
PARAM L(HCP_A3,FE:C,VA;0) 298.15 -22126; 6000 N REF3 !
$PARAM L(HCP_A3,FE:C,VA;0) 298.15 -17335; 6000 N TCFE3 !
PARAM L(HCP_A3,CR,FE:VA;0) 298.15 +10833-7.477*T; 6000 N REF4 !

$***** SIGMA (Solution) *****
PHASE SIGMA % 3 8 4 18 !
CONST SIGMA : FE
              : CR
              : CR,FE : !

PARAM G(SIGMA,FE:CR:CR;0) 298.15 +8*GFEFCC#+22*GHSERCR#
      +92300-95.96*T+GPSIG1#; 6000 N REF4 !
PARAM G(SIGMA,FE:CR:FE;0) 298.15 +8*GFEFCC#+4*GHSERCR#+18*GHSERFE#
      +117300-95.96*T+GPSIG2#; 6000 N REF4 !

$***** CEMENTITE (Solution) *****
PHASE CEMENTITE % 2 3 1 !
CONST CEMENTITE : CR,FE%
                  : C : !

PARAM G(CEMENTITE,CR:C;0) 298.15 +3*GHSERCR#+GHSECC#
      -48000-9.2888*T; 6000 N REF2 !
PARAM G(CEMENTITE,FE:C;0) 298.15 +GFECM#; 6000 N REF3 !

PARAM L(CEMENTITE,CR,FE:C;0) 298.15 +25278-17.5*T; 6000 N REF2 !

$***** M3C2 (Compound) *****
PHASE M3C2 % 2 3 2 !
CONST M3C2 : CR
            : C : !

PARAM G(M3C2,CR:C;0) 298.15 +GCRM3C2#; 6000 N REF2 !

$***** M7C3 (Solution) *****
PHASE M7C3 % 2 7 3 !
CONST M7C3 : CR%,FE%
            : C : !

PARAM G(M7C3,CR:C;0) 298.15 +GCRM7C3#; 6000 N REF2 !
PARAM G(M7C3,FE:C;0) 298.15 +7*GHSERFE#+3*GHSECC#
      +75000-48.2168*T; 6000 N REF2 !

PARAM L(M7C3,CR,FE:C;0) 298.15 -4520-10*T; 6000 N REF2 !

$***** M23C6 (Solution) *****
PHASE M23C6 % 3 20 3 6 !
CONST M23C6 : CR%,FE%
              : CR%,FE%
              : C : !

PARAM G(M23C6,CR:CR:C;0) 298.15 +GCRM23C6#; 6000 N REF5 !
PARAM G(M23C6,FE:CR:C;0) 298.15 +.1304348*GCRM23C6#
      +.8695652*GFEM23C6#; 6000 N REF5 !
PARAM G(M23C6,CR:FE:C;0) 298.15 +.8695652*GCRM23C6#
      +.1304348*GFEM23C6#; 6000 N REF5 !
PARAM G(M23C6,FE:FE:C;0) 298.15 +GFEM23C6#; 6000 N REF5 !

```