

## Thermocalc Scripts

The information below pertains to using scripts in THERMOCALC (for scripts in drawpd, see the drawpd documentation). This list is an updated version of the list given in the course CD documentation originally compiled by Roger Powell and updated now by Richard White.

Clicking a script below will take you to a more in-depth explanation of what the script does and what commands can follow it.

### Overview

Scripting allows simple customisation of the running of THERMOCALC. This is a good idea because each time you run the program you would otherwise be met with a barrage of questions many of which you might know in advance are not relevant or that can be set ahead of time. It is usually possible to run THERMOCALC without using scripts, but using them can make your life much easier. In addition some facilities are only available via scripts.

There are two places that scripts are read by THERMOCALC :

1. in a separate file called tc-prefs.txt: these scripts control global aspects of running THERMOCALC. Scripts here will affect all runs of THERMOCALC with different datafiles,
2. in the scriptfile (or the second section of a datafile in older THERMOCALC versions): these local scripts control the way THERMOCALC runs that particular axfile.

Each of the controls which can be set by scripts are set to a default in THERMOCALC. All that happens with the provision of scripts in tc-prefs or in a datafile is that the default is overridden. Each time it is run, THERMOCALC reads these scripts, and these override the defaults; THERMOCALC does not store or remember previously read scripts.

The basic script format involves a single line of file, starting with a “keyword”. If you don’t get the keyword right (ie if THERMOCALC doesn’t recognise it), then the line is ignored and (generally) there will be a line of output at the beginning of a run that this has happened “ script unrecognised” (but THERMOCALC won’t bomb).

The simplest form of script involves a “keyword” followed by one of “yes”, “no” or “ask”. “yes” corresponds to some action which will definitely be undertaken, and “no” corresponds to the action definitely not being undertaken. With “ask”, you will be prompted by THERMOCALC, while the program is running, to determine if the action should be undertaken. In some cases “no” gets substituted for “ask” with no prompt allowing the meaning of “ask” to be clarified. If one of “yes”, “no” or “ask” is omitted, then “yes” is assumed.

More complex scripts allow additional information to be provided to THERMOCALC. Remember that if THERMOCALC doesn’t understand what you have

tried to do with a script, there is no problem, but that the intended action or inaction will not occur. This will normally be obvious in the way THERMOCALC runs! As each script is a single line of datafile, one script which doesn't work will not affect succeeding scripts. (In the following, the forms of scripts are in quotes—don't include the quotes when you use the scripts)

Below is a list of scripts as at Dec 2007 for THERMOCALC 3.30. While many scripts here work in earlier versions, not all of them will. The links in each script will take you to a more in-depth explanation of what the script does, what script suffixes can be used and how to use each script. The scripts are separated into two groups, being those for the preferences file and those for the script file. The scripts serve a range of purposes including telling THERMOCALC what files to use, setting what types of calculations to run, what types of output to produce, what information to include in that output and controlling things such as minerals to use, P-T windows etc. A few scripts that have been recently superseded or no longer work in THERMOCALC 3.30 but are relevant to recent versions such as 3.23-3.26 are included

For calculating particular types of diagrams, ready-made script files with the set of scripts required are available for download in the [datafiles](#) page.

### **Scripting in the preferences file**

The most useful scripts which can be used in the prefs are:

[dataset](#) – specifying which version of the internally consistent dataset to use (currently 55, meaning that tc-ds55.txt is used)

[scriptfile](#) - specifying which script file suffix to use (replaces datasuffix)

[setpagewidth](#) - allows the width of the calculation window to be set (80 for the pc black window; whatever size is used for the Terminal window on the Mac)

[dontwrap](#) - whether line wrap occurs in screen output (for specified pagewidth)

[calcmode](#) - specifying a calculation mode for all runs

[strictsyntax](#) – better ax code checking, but involves a strict naming convention for variables

main pre 3.30 scripts, equivalent to some of above

[datasuffix](#) - specifying a datafile suffix

[pdatasuffix](#) - specifying a suffix on tc<dataset> eg ds55

### **Scripting in mode 1 datafiles**

Scripts for phase diagram calculations that may appear in script files. Clicking on a script takes you to a more in-depth explanation of the script and what suffixes can/must follow it.

## Defining the a-x file to use

[axfile](#) - defining the a-x file to use

## Related to calculation *PT* 'window'

[setdefPwindow](#) - defining the default *P* window for calculations

[setdefTwindow](#) - defining the default *T* window for calculations

[setPwindow](#) - defining the *P* range for calculations

[setTwindow](#) - defining the *T* range for calculations

[calctatp](#) - whether to calculate *T* at given *P* for *PT*-lines

[moreprec](#) - specifying more output precision (now standard in version 3.30)

[calcsdnle](#) - whether to perform uncertainty propagation on calculations

## Related to phases involved in calculations

[omit or ignore](#) - specifying phases to ignore this run

[fluidpresent](#) - whether H<sub>2</sub>O or CO<sub>2</sub> in the datafile means fluid present

[fluidexcess](#) - whether fluid is "in excess"

[setexcess](#) - defining other "in excess" phases

[exbuff](#) - allows external buffering calculations involving H<sub>2</sub>O-CO<sub>2</sub> fluids

[seta](#) - whether to set activities in calculations

[setmu](#) - whether to set chemical potentials in calculations

[usecalcq](#) - whether to pre-calculate the state of order in order-disorder phases

[samecoding](#) - whether to line up output for minerals with the same a-x code for easy inspection

## Related to calculating pseudosections and setting bulk rock composition

[pseudosection](#) - whether to generate pseudosection information

[setbulk](#) - defining bulk composition(s) for pseudosections

[setmodeiso](#) - allows the setting of mode isopleths

[zeromodeiso](#) - allows the setting of zero mode isopleths

[setiso](#) - whether to calculate composition isopleths in calculations

[calcg](#) - Gibbs energy of equilibrium calculated

[dogmin](#) - allows Gibbs energy minimization

[rbi](#) - allows a matrix of calculated mineral modes and mineral compositions to be used as the bulk rock composition

[calcah2o](#) - whether to calculate the activity of water

[calcmu](#) - whether to calculate the chemical potential of each component

## Output for drawpd

[drawpd](#) - calculated coordinates into the file called tc-blah-dr, where blah is the name given in the scriptfile script (or prompted)

## Related to calculating projections

[project](#) - whether to project phases onto a compatibility diagram

[setproject](#) - defining the compatibility diagram (projection plane)

[projcomp](#) - whether to project bulk comps onto the compatibility diagram

## Optional output in the log file

[printbulkinfo](#) - whether to include a bulk composition matrix in the logfile

[printguessform](#) - whether to include the output mineral compositions in starting guess form in the logfile (superceded by [printxyz](#))

[printxyz](#) – whether to include the output mineral compositions in starting guess form in the logfile

[incthermo](#) - whether to include tables of thermo data in the logfile

[incax](#) - whether to include the *a-x* relationships in the logfile

[incnler](#) - whether to include the set of reactions in the logfile

[lsqds](#) –allows activities of end-members rather than mineral comps to be output for calculations

### **to produce Mathematica output (mainly for wizards)**

[smath](#) - calculated *PT* coordinates into 'other' file

[xinsmath](#) - calculated *PT* and *x* coordinates into 'other' file

[tabsmath](#) - calculated *PT* coordinates into 'other' file in tab-separated form

[cmath](#) - compatibility diagram coordinates into 'other' file

[txmath](#) - *Tx/Px* pseudosection coordinates into 'other' file

### **scripts for avPT (mode 2)**

[defsdaparams](#) – for setting activity uncertainty defaults

[usedefsda](#) – using the defaults

[justrelu](#) - just relative uncertainties (as in the thermobarometry paper)

[setrockname](#) – allows the results to be labelled with the rock name

[diagnostics](#) – to always see the diagnostics

### **scripts for calculations not involving (solid) solutions (mode 3)**

[setmu](#) – allows you to set the chemical potentials of components

[seta](#) / [setlna](#) / [setloga](#) – allows you to set activities of components

### **Scripts Explanations**

Below is a list of usable scripts as at Dec 2007 for THERMOCALC 3.30. While most scripts here are usable in earlier versions, not all of them will be (so using THERMOCALC 3.30 is recommended). The links in each script will take you to a more in-depth explanation of what the script does, what script suffixes can be used and how to use each script. The scripts are separated into two groups, being those for the preferences file and those for the script file. The scripts serve a range of purposes including telling thermocalc what files to use, setting what types of calculations to run, what types of output to produce, what information to include in that output and controlling things such as minerals to use, P-T windows etc.

This list can be downloaded in pdf format [here](#)

For a description of the file structure for input and output files [click here](#)

### **Scripting in the preferences file**

**dataset** (*for version 3.30 onward*) The file with the prefix/root tc-ds is the internally consistent dataset (eg tc-ds55) that is to be used in the calculations. On most cases, this does not change very often (eg tc-ds55 is from Nov. 2003). For calculations, the dataset to be used is specified in the tc-prefs file. Forms

“dataset blah” where “blah” is the text that follows the “tcds” (eg “dataset 55” to use tc-ds55)

**scriptfile** (*for version 3.30 onward*) This script is available if you are spending a lot of time working with just one scriptfile; it allows you to specify this file in tc-prefs, via the suffix to the root of the datafile name. Having done this, it means that you can only work with this file (until you change the script, or remove it). This root is either the default, ‘tc-’, or whatever is specified in the script datafilename. If “scriptfile” is not present in tc-prefs, then the suffix to ‘tc-<name>-o’ (or whatever is specified in the script datafilename) is prompted for when you run thermocalc. Form:

“scriptfile blah”            scriptfile used is tc-blah (Win: tc-blah.txt)

Default: datafile name suffix will be prompted for

**setpagewidth** allows the width of the calculation window to be set (80 for the pc black window; whatever size is used for the Terminal window on the Mac)

**dontwrap** Screen output on smaller screens (or longer lines of output) can look messy if a line is wrapped onto the following one. This script allows user control over this. With “outdialog yes”, a # is placed at the end of a line which would otherwise wrap onto the following line. The full line of course does appear in the output file. Use in conjunction with “dontaskadjust”, to control the width of window on which the wrapping is not done. Forms:

“dontwrap yes”

or

“dontwrap”                    don’t wrap output on screen

“dontwrap no” allow wrap of output on screen

Default: “dontwrap yes”.

**calcmode** If you are only doing one type of calculation, say only average PT calculations, then you can specify this with this script (and save yourself always replying in this way to a prompt). Form:

“calcmode -1”            calculation mode prompted for (this is the equivalent of the script not occurring in th prefs)

“calcmode 0”            data table mode (this would not be a usual setting)

“calcmode 1”            phase diagram calculations involving phases which are (solid) solutions

“calcmode 2”            rock calculations (ie average PT calculations)

“calcmode 3”            calculations involving all the reactions between a set of end-members (for phase diagrams not involving (solid) solutions and debugging average PT calculations)

Default: calculation mode prompted for

**strictsyntax** allows better ax code checking, but requires a strict naming convention for all variables.

“strictsyntax yes”

or

“strictsyntax” strict syntax checked for

“strictsyntax no” allows a less rigid ax coding

Default: strictsyntax yes

**datasuffix** (*for pre version 3.30*) This script is available if you are spending a lot of time working with just one datafile; it allows you to specify this file in th prefs, via the suffix to the root of the datafile name. Having done this, it means that you can only work with this file (until you change the script, or remove it). This root is either the default, ‘th d’, or ‘tcd’ or whatever is specified in the script datafilename. If “datasuffix” is not present in th prefs, then the suffix to ‘th d’ (or whatever is specified in the script datafilename) is prompted for when you run thermocalc. Form:

“datasuffix blah” datafile used is th dblah (Win: thblah.txt)

Default: datafile name suffix will be prompted for

**Pdatasuffix** (*pre version 3.30*) The pdata file contain the thermodynamic data which thermocalc uses for all its calculations. As the pdata files are updated, the results of calculations change (if usually only slightly). Sometimes it is useful to keep older pdata files, so that original calculations can be reproduced. This script allows several generations of pdata files to be kept together, using different suffices on the pdata filenames, and the choice of which ones are used being controlled by a tcprefs script. Forms:

“pdatasuffix blah” pdata file will be expected to have a name like tc-ds55 (dataset 5.5)

Default: files are named tc-ds55

## Scripting in mode 1 datafiles

**axfile** This script sets the a-x file to be used in the calculations. A-x files are typically named with the chemical system (eg “tc-KFMASH”) and contain the activity-composition relationships, though they can have any name you choose. Form:

“axfile <name>”

“axfile ask”

Default: "axfile ask"

**setdefPwindow** This script sets a default P window to be used if the user hits return (ie inputs nothing) at a prompt for a P (or PT) range. The 2 numbers "defplow" and "defphigh" defining the window must be given. Form:

"setdefPwindow defplow defphigh"  
or  
"setdefPwindow yes defplow defphigh"      the given numbers define the default P window;  
"setdefPwindow ask defplow defphigh"      you are prompted to determine if the specified window is to be used;  
"setdefPwindow no defplow defphigh"      the specified window is ignored

eg "setdefPwindow 2 8"

Default: 0.1 to 20 kbar

**setdefTwindow** This script sets a default T window to be used if the user hits return (ie inputs nothing) at a prompt for a T (or PT) range. The 2 numbers "deftlow" and "defthigh" defining the window must be given. Form:

"setdefTwindow deftlow defthigh"  
or  
"setdefTwindow yes deftlow defthigh"      the given numbers define the default T window;  
"setdefTwindow ask deftlow defthigh"      you are prompted to determine if the specified window is to be used;  
"setdefTwindow no deftlow defthigh"      the specified window is ignored

eg "setdefPwindow 400 800"

Default: 200 to 1000°C

**setPwindow** This script allows the P window within which equilibria are considered to be specified in the datafile. The 2 numbers "plow" and "phigh" defining the window must be given. With these "phigh"  $\geq$  "plow", with, for example, equality not being allowed if the calculated equilibria are invariant. If equality is specified, and is not allowed, thermocalc prompts for a proper range. For calculations at a fixed P (eg for a T-x pseudosection) the script can be followed by only a single number or the same number twice Form:

"setPwindow plow phigh"  
or  
"setwindow yes plow phigh"      the given numbers define the P window;  
"setPwindow ask plow phigh"      you are prompted to determine if the specified window is to be used;  
"setPwindow no plow phigh"      the specified window is ignored

eg "setPwindow 2 8" to calculate in this range or "setPwindow 8 8" to calculate at fixed P

Default: prompt for P window.

**setTwindow** This script allows the T window within which equilibria are considered to be specified in the datafile. The 2 numbers “tlow” and “thigh” defining the window must be given. With these “thigh” ≥ “tlow”, with, for example, equality not being allowed if the calculated equilibria are invariant. If equality is specified, and is not allowed, thermocalc prompts for a proper range. For calculations at a fixed T (eg for a P-x pseudosection) the script can be followed by only a single number or the same number twice Form:

“setTwindow tlow thigh”

or

“setwindow yes tlow thigh”

“setTwindow ask tlow thigh”

the given numbers define the T window;  
you are prompted to determine if the specified window is to be used;

“setTwindow no tlow thigh”

the specified window is ignored

eg “setTwindow 200 800” to calculate in this range or “setTwindow 600 600” to calculate at fixed T

Default: prompt for T window.

**calctatp** With a PT window (see setwindow), and considering univariant equilibria (ie reaction lines) in a PT window, there is a choice of whether T should be calculated at a series of P, or P calculated at a series of T. For example, for reactions with a very small  $dP/dT$ , it is easier to calculate P at T. This script allows this choice to be made. (If for the window  $p_{low}=p_{high}$  or  $t_{low}=t_{high}$ , then the choice does not arise). Form:

“calctatp”

or

“calctatp yes”

“calctatp ask”

“calctatp no”

always calculate T at P;  
you are prompted for the choice;  
always calculate P at T

Default: you are prompted for the choice

**moreprec (defunct in version 3.30)** Allows an extra decimal place in the output of PT and x values. This is useful if equilibria are closely spaced - note, however, that there is no implication that the numbers are somehow more accurate when this switch is on. Form:

“moreprec”

or

“moreprec yes”

“moreprec ask”

“moreprec no”

an extra decimal place is used in the output;  
you are prompted whether an extra decimal place should be used in the output;  
no extra decimal place in the output.

Default: “moreprec no”

**calcsdnle** Allows uncertainty propagation calculations to be switched on or off.

Note the uncertainties given are 1sigma, but should be quoted at 2sigma (95% confidence). Form:

“calcsdnle”

or

“calcsdnle yes” uncertainty propagation calculations will be performed;

“calcsdnle ask” you are prompted whether uncertainty propagation calculations will be performed;

“calcsdnle no” no uncertainty propagation

Default: “calcsdnle no”

**omit** or **ignore** Allows phases for which information is provided in the first part of a datafile to be ignored in using the datafile. This circumvents the need for lengthy lists of phases to be omitted at the ‘which phases’ prompt when running thermocalc or alternatively the cutting and pasting of phase info from the first part to the last (storage) part of datafiles. Form:

“ignore phase1 phase2 etc”

or

“ignore yes phase1 phase2 etc” phase1, phase2, etc, which are present in the first part of the datafile, are to be ignored in the calculations

“ignore ask phase1 phase2 etc” you are prompted whether these phases are to be ignored;

“ignore no anything” no phases ignored

“ignore \_\_\_\_\_” nothing following the script means no phases ignored

Default: nothing ignored

**fluidpresent** In phase diagram calculations involving H<sub>2</sub>O or CO<sub>2</sub> (but not H<sub>2</sub>O and CO<sub>2</sub>, see below), these end-members might be included in a system to be used in fluid-absent equilibria (say with a(H<sub>2</sub>O) set at particular values). This script is used to communicate that a system may have H<sub>2</sub>O or CO<sub>2</sub> present as phases. If a fluid phase with H<sub>2</sub>O and CO<sub>2</sub> is in the system, it is coded in the datafile as a solution, much as solid solutions are, and mode 1 should be used. In this case fluidpresent is automatically yes, and does not need to be scripted. Form:

“fluidpresent”

or

“fluidpresent yes“ H<sub>2</sub>O or CO<sub>2</sub> is present as a phase;

“fluidpresent ask” prompt - this is thermocalc normal behaviour in the absence of the script;

“fluidpresent no” H<sub>2</sub>O or CO<sub>2</sub> cannot be present as a phase

Default: “fluidpresent ask”

**fluidexcess** In calculations, it is often convenient to consider one or more

phases to be “in excess”, to reduce the total number of equilibria to be considered, and also to reduce the effective dimension of the system so that compatibility diagrams can be drawn (eg the 6-component system KFMASH is reduced to the effective ternary, AFM, by having muscovite, quartz and H<sub>2</sub>O in excess). The fluidexcess script allows fluid in excess to be specified. Form:

“fluidexcess”  
or  
“fluidexcess yes” fluid is in excess;  
“fluidexcess ask” prompt - this is thermocalc normal behaviour in the absence of the script;  
“fluidexcess no” fluid not in excess

Default: “fluidexcess ask”

**setexcess** In calculations, it is often convenient to consider one or more phases to be “in excess”, to reduce the total number of equilibria to be considered, and also to reduce the effective dimension of the system so that compatibility diagrams can be drawn (eg the 6-component system KFMASH is reduced to the effective ternary, AFM, by having muscovite, quartz and H<sub>2</sub>O in excess). The setexcess script allows phases other than fluid to be considered to be in excess (see fluidexcess). Form:

“setexcess ph1 ph2 ph3 ...”  
or  
“setexcess yes ph1 ph2 ph3 ...” the phases ph1, ph2, and so on are taken to be in excess (above eg “setexcess mu q”);  
“setexcess ask ph1 ph2 ph3 ...” prompt to determine if the phases ph1, ph2, and so on are taken to be in excess;  
“setexcess no ph1 ph2 ph3 ...” nothing in excess

Default: prompt for in excess phases.

**exbuff** Allows external buffering calculations involving H<sub>2</sub>O-CO<sub>2</sub> fluids to be done. For example in calculating a T-x(CO<sub>2</sub>), in combination with “setiso xCO<sub>2</sub>”, it allows pseudosections to be drawn. Form:

“exbuff”  
or  
“exbuff yes” external buffering is activated;  
“exbuff no” not external buffering.

Default: “exbuff no”

**seta** In phase diagram calculations, it is sometimes important to see what the consequence is of varying the activity of an end-member, for example H<sub>2</sub>O (under granulite facies conditions). This script allows this to be done. Form:

“seta”  
or  
“seta yes” you will be prompted for the name of the end-member whose activity will be fixed;

"seta em"  
 or  
 "seta yes em"      the activity of end-member, em, will be fixed (eg "setaH<sub>2</sub>O");  
 "seta ask"          you will be prompted to determine if you wish to fix an activity;  
 "seta ask em"      you will be prompted to determine if you wish to fix the  
                                  activity of end-member, em (eg "seta ask H<sub>2</sub>O");  
 "seta no anything"      you do not wish to fix an activity.

Default: "seta no"

**setmu** In phase diagram calculations, it is sometimes important to see what the consequence is of varying the chemical potential of a component, for example K<sub>2</sub>O (in orthoamphibole assemblages). This script allows this to be done. This script is also used for construction of  $\mu$ - $\mu$  diagrams via setting a range of  $\mu$ 's to calculate lines Form:

"setmu"  
 or  
 "setmu yes"              you will be prompted for the name of a component whose  
                                  chemical potential will be fixed and later, the chemical  
                                  potential value or range of values and an increment;  
  
 "setmu comp"  
 or  
 "setmu yes comp"      the chemical potential of component, comp, will be fixed  
                                  (eg "setmu Na<sub>2</sub>O");  
  
 "setmu comp1 num1"  
 or  
 "setmu yes comp1 num1"      the chemical potential of component, comp, will be fixed  
                                  at the defined value (eg "setmu Na<sub>2</sub>O -960"). More than  
                                  one component may have a fixed mu  
  
                                  (eg "setmu yes K<sub>2</sub>O -939.6960 Al<sub>2</sub>O<sub>3</sub> -1751.0310" )  
  
 "setmu ask"              you will be prompted to determine if you wish to fix a  
                                  chemical potential;  
 "setmu ask comp"      you will be prompted to determine if you wish to fix the  
                                  chemical potential of comp (eg "setmu ask Na<sub>2</sub>O");  
 "setmu no anything"      you do not wish to fix a chemical potential.

Default: "setmu no"

**usecalcq** This script prompts thermocalc to solve the state of order in order-disorder minerals prior to solving for the specified equilibria. The state of order calculated is then used as a starting guess. The state of order is solve for the P-T condition in the middle of the P-T window specified. Form:

"usecalcq"  
 or  
 "usecalcq yes" the state of order pre-calculated for order-disorder phases  
 "usecalcq ask"      you will be prompted whether you wish the state of order to be  
                                  pre-calculated  
 "usecalcq no"      the state of order pre-calculated

Default: "usecalcq yes"

**samecoding** This script causes output to be lined up for visual comparison. All the samecoding phases need to be at the beginning of the axfile. (This is a precursor to just putting the a-x relations in once each for amph, cpx etc). This is currently being debugged - if you are having problems with it, just don't use it (and maybe report to RP what happens). Form:

"samecoding min1 min2 min3"                      causes output to be lined up for visual comparison

eg to use same coding for coexisting amphiboles and coexisting cpx

samecoding act hb gl

samecoding jd o di

on separate lines like this.

**readxyz (version 3.30 and above)** This script followed by a composition variable and a numeric value allows the starting guesses to be set in the scripts file. Doing so makes these starting guesses overwrite those in the a-x file for each calculation. A thermocalc-readable list can be printed in the tc-log file using the "printxyz" script (see above). This is an easy way to change the starting guesses without having to open the a-x file Form:

"readxyz var num"    this sets this number as the starting guess for the chosen variable (eg "readxyz x(bi) 0.4284")

Default: "script not present"

**pseudosection** In phase diagram calculations, it is very informative to know if a particular equilibrium is "seen" by a particular bulk composition. Phase diagrams, for example PT diagrams, drawn to show just those equilibria "seen" by a particular bulk composition are referred to as pseudosections. This script allows pseudosection information to be collected. Typically the bulk composition is set using the "setbulk" or "rbi" script but can be input manually at each calculation. Form:

"pseudosection"

or

"pseudosection yes"

you are prompted for the bulk composition(s) you want to collect pseudosection information for;

"pseudosection ask"

you are prompted concerning whether you want to collect pseudosection information; if you reply yes, you are prompted for the bulk composition(s) you want to collect pseudosection information for; if you reply no, pseudosection information is not collected;

"pseudosection no"

pseudosection information is not collected.

Default: "pseudosection ask"

**setbulk** If pseudosection is yes, then setbulk allows one or two bulk compositions (in mole %) to be scripted. The setbulk script is followed by the bulk composition in the system defined by the a-x models. Each bulk composition has the components in a specific order. The simplest way to get this order correct is to run thermocalc with the datafile of interest, with "pseudosection yes" and without the setbulk script. The bulk composition is then prompted for in the correct order. The bulk compositions input do not have to sum to 100, as this normalization is done by thermocalc. Each component in the system (including those only in in-excess phases) must be defined with the exception of fluid. Forms:

"setbulk yes com1 com2 com3....."

or

"setbulk com1 com2 com3....."

"setbulk com1 com2 com3....."

"setbulk no com1 com2 com3....."

use this bulk composition (com = component)  
prompt to ask the user whether this bulk  
composition is to be used

this bulk composition is not to be used, and the  
user will be prompted for the bulk composition  
(useful for temporarily "switching off" the script)

An example of how this script looks in a typical P-T pseudosection file is below:

```
% -----  
% kfmash composition  
% -----  
%          H2O  SiO2  Al2O3  MgO  FeO  K2O  
setbulk yes 6.56 68.76 9.87  4.01 7.64 3.16  
% -----
```

An example of how this script looks in a typical P-x or T-x pseudosection file is below:

```
% -----  
% kfmash composition P/T-xFe  
% -----  
%          H2O  SiO2  Al2O3  MgO  FeO  K2O  
setbulk    6.56 68.76 9.87  11.65 0.00 3.16  
setbulk yes 6.56 68.76 9.87  0.00 11.65 3.16  
% -----
```

**setmodeiso** In pseudosection calculations, a useful facility is to be able to fix the value of modes of phases, and calculate all the other composition and mode variables of the phases in an equilibrium. This facility allows the position of mode isopleths on a PT pseudosection to be calculated. In particular, it allows (in conjunction with the "zeromodeiso" script) the boundaries between fields on a pseudosection to be calculated. Regardless of the variance of the field in a PT pseudosection, with one mode fixed, the equilibrium is effectively univariant (ie defines a PT line), and with two modes fixed, the equilibrium is effectively invariant (ie defines a PT point). Form:

"setmodeiso"

or

"setmodeiso yes" you will be prompted for the names of the composition

variables to be fixed;

“setmodeiso ph”  
or  
“setmodeiso yes ph” the mode of the phase, ph, is to be fixed, and the values are prompted for (eg “setmodeiso bi”);

“setmodeiso ph isolow isohigh isoinc”  
or  
“setmodeiso yes ph isolow isohigh isoinc” the mode of the phase, ph, is to have values running from isolow through to isohigh with an interval, isoinc (eg “setiso bi 0.1 0.6 0.1”);

“setmodeiso ask” you will be prompted to determine if you wish to fix any composition variables, then prompted for the details;

“setmodeiso ask ph” you will be prompted to determine if you wish to fix the mode of the phase, ph (eg “setmodeiso ask bi”);

“setmodeiso ask ph isolow isohigh isoinc” you will be prompted to determine if you wish to fix the composition variable, ph, and to have values running from isolow through to isohigh with an interval, isoinc;

“setmodeiso no anything” → you do not wish to fix any modes.

Default: “setmodeiso ask”

**zeromodeiso** This qualifies “setmodeiso yes”, making the setting of modes always involve zero values. This facility is central to calculating the lines and points that define field boundaries in PT and Tx pseudosections. Form:

“zeromodeiso”  
or  
“zeromodeiso yes” allows zero mode isopleths to be set;  
“zeromodeiso ask” you are prompted whether you want zero mode isopleths to be set;

“zeromodeiso no” zero mode isopleths are not set

Default: “zeromodeiso ask”

**setiso** In phase diagram calculations, an important facility is to be able to fix the value of any compositional variable, and calculate all the other composition variables of the phases in an equilibrium. This facility allows the position of isopleths on a PT diagram to be calculated, as well as the edges of trivariant fields on compatibility diagrams. For example, in AFM (in KFMASH), the compositions of, say, coexisting biotite and chlorite can be calculated via fixing  $x(\text{Fe})$  in the biotite (or chlorite). A major application concerns Tx(CO<sub>2</sub>) (and P-x(CO<sub>2</sub>)) diagrams: by fixing  $x(\text{CO}_2)$  in the fluid and calculating equilibria such diagrams can be calculated. Care is needed in the specification of variance for such calculations. Remember that univariant lines on T-x(CO<sub>2</sub>) diagrams are actually divariant on a PT diagram. Thus with  $x(\text{CO}_2)$  to be specified, the required variance when prompted to calculate univariant lines on a T-x(CO<sub>2</sub>) is trivariant. In general, for each composition variable fixed, the (effective)

variance is decreased by one. Form:

“setiso”

or

“setiso yes”                    you will be prompted for the names of the composition variables to be fixed;

“setiso var”

or

“setiso yes var”                the composition variable, var, is to be fixed, and the values are prompted for (eg “setiso x(bi)“);

“setiso var isolow isohigh isoinc”

or

“setiso yes var isolow isohigh isoinc”                the composition variable, var, is to have values running from isolow through to isohigh with an interval, isoinc (eg “setiso x(bi) 0.1 0.6 0.1“);

“setiso ask”                    you will be prompted to determine if you wish to fix any composition variables;

“setiso ask var”                you will be prompted to determine if you wish to fix the compositional variable, var (eg “setiso ask x(bi)“);

“setiso ask var isolow isohigh isoinc”                you will be prompted to determine if you wish to fix the composition variable, var, to have values running from isolow through to isohigh with an interval, isoinc;

“setiso no anything”    you do not wish to fix any composition variables.

Default: “setiso ask”

**calcg** The Gibbs energy of each equilibrium is calculated. Form:

“calcg”

or

“calcg yes”    calculate Gibbs energy;

“calcg ask”    prompt for calculation of Gibbs energy;

“calcg no”    do not calculate Gibbs energy

Default: “calcg no”

**dogmin** Allows Gibbs energy minimisation calculations: for a specified bulk composition all of the equilibria from divariant up to a specified variance are calculated and the one, at each PT, that has the lowest Gibbs energy is the stable one. This can be useful for kick-starting the calculation of a pseudosection, but in large chemical systems can be very slow. Form:

“dogmin”

or

“dogmin yes”    do Gibbs energy minimisation;

“dogmin ask”    prompt for Gibbs energy minimisation;

“dogmin no”    Gibbs energy minimisation not done

Default: “dogmin no”

**rbi** This script allows the bulk rock composition to be defined via a combination

of mineral compositions and modes. This can be useful for adding/removing a proportion of a given phase (eg to model melt loss). A readable table of mineral compositions and modes can be printed into the tc-log file using the script "printbulkinfo" Form:

the table of mineral compositions will look something like this:

```
% =====
%          cd          g          liq          ksp          q
rbi yes  0.128725  0.142898  0.590856          0  0.137521

%          H2O          SiO2          Al2O3          CaO          MgO          FeO          K2O          Na2O
rbi      0.355525  5.000000  2.000000          0  1.245074  0.754926          0          0 % cd
rbi      0  3.000000  1.000000  0.021669  0.750962  2.227370          0          0 % g
rbi      0.320534  2.252573  0.216288  0.012426  0.020572  0.070052  0.149223  0.029450 % liq
rbi      0  2.993191  0.503404  0.006809          0          0  0.458251  0.038344 % ksp
rbi      0  1.000000          0          0          0          0          0          0 % q
% =====
```

**calcah2o** This script allows the activity of water to be calculated for each calculation. This is useful to investigate variations in aH<sub>2</sub>O around a diagram Form:

```
"calcah2o"
or
"calcah2o yes" the activity of water will be calculated
"calcah2o ask" you will be prompted whether you want the activity of water to be
                calculated
"calcah2o no"  the activity of water will not be calculated
```

Default: "calcah2o no"

**calcmu** This script allows the chemical potential of each component to be calculated for each calculation. This is used for construction of  $\mu$ - $\mu$  diagrams. The chemical potential values will also be printed in the tc-<filename>-dr.txt file. Form:

```
"calcmu"
or
"calcmu yes"  the chemical potentials will be calculated
"calcmu ask"  you will be prompted whether you want the chemical potentials
                to be calculated
"calcmu no"   the chemical potentials will not be calculated
```

Default: "calcah2o no"

**drawpd** If "makeo yes" is in the preferences file, indicating that a 'other' file is to be made (whose default name is 'tc-<filename>-dr.txt'), this causes coordinate information from the calculations to be put in it in a format that is readable by the drawpd software. Form:

```
"drawpd"
or
"drawpd yes" put coordinate info in 'other' file;
"drawpd no"  nothing put
```

Default: "drawpd no"

**project** In representing the results of phase diagram calculations, it is often convenient to be able to project the calculated compositions of the phases into a compatibility diagram. This script allows this. Form:

"project"  
or  
"project yes" you are prompted for the definition of the compatibility diagram you want to use;  
"project ask" you are prompted concerning whether you want to project the phase diagram information; if you reply yes, you are prompted for the definition of the compatibility diagram you want to use; if you reply no then projection is not performed;  
"project no" projection is not performed.

Default: "project ask"

**setproject** If project is yes, then setproject allows the definition of the compatibility diagram (or projection plane) to be scripted, at least for compatibility diagrams which have oxides as their corners. So for example, for KFMASH projected from muscovite, quartz and H<sub>2</sub>O (ie having muscovite, quartz and H<sub>2</sub>O in excess), Al<sub>2</sub>O<sub>3</sub>, FeO and MgO (ie AFM) define the corners of the usually used compatibility diagram. The oxides are specified in terms of their position in the list of oxides given by thermocalc when this same information

is prompted for. The simplest way to get this script correct is to run thermocalc with the datafile of interest, with "project yes" and without the setproject script. Forms:

"setproject yes 1 3 2"  
or  
"setproject 1 3 2" use 1st, 3rd and 2nd component  
"setproject ask 1 3 2" prompt to ask the user whether this is to be used  
"setproject no 1 3 2" this is not to be used, and the user will be prompted for the  
projection plane (useful for temporarily "switching off" the script)

below is the setproject script for a standard AFM diagram

```
% -----  
setproject      A          1    0    0    0  
setproject      F          0    0    1    0  
setproject yes  M          0    1    0    0  
% -----
```

**projcomp** In phase diagram calculations involving projection and the collection of pseudosection information, particularly when there is projection from variable composition phases, it can be useful to see where the bulk composition(s) plot on the compatibility diagram (projection plane). This script allows this, as long as project is yes and pseudo is yes. A bulk composition is only projected if the



composition results from each calculated point to be printed in the tc-log file as a thermocalc-readable list of mineral compositions in the form of the composition variables for each phase. This can then be copied and pasted into the script file as the starting guesses. Doing so makes these starting guesses overwrite those in the a-x file. Form:

“printxyz”

or

“printxyz yes”           the mineral composition list is printed in the tc-log file

“printxyz ask”           you will be prompted whether you want this list printed

“printxyz no”           the mineral composition list is not printed in the tc-log file

the list will look something like this:

```
% calculated at P = 6.9; T = 800
readxyz q(L)           0.1446
readxyz fsp(L)         0.3061
readxyz na(L)          0.6839
readxyz an(L)          0.01097
readxyz ol(L)          0.009983
readxyz x(L)           0.7444
readxyz h2o(L)         0.5197
readxyz x(bi)          0.4284
readxyz y(bi)          0.2536
readxyz f(bi)          0.03740
readxyz t(bi)          0.1263
readxyz Q(bi)          0.2588
readxyz x(g)           0.7238
readxyz z(g)           0.02970
readxyz f(g)           0.005363
readxyz ca(pl)         0.3011
readxyz k(pl)          0.03034
```

Default: “printxyz no”

**incthermo** In phase diagram calculations, there is the option of a table of the thermodynamic data being included in the logfile. This script allows it to happen. Usually there is not much point in including the table, given that it takes a little time to print. One situation it might be useful, though, is when the DQF facility or the dependent end-member facility is used, in which case the data for the constructed end-member may be of interest. Form:

“incthermo”

or

“incthermo yes”       include the table;

“incthermo no”        table not included

Default: table not included

**incax** In phase diagram calculations, there is the option of a summary of the activity-composition relationships in the datafile being included in the logfile. This script controls this. Given that coding the activity-composition relationships provides lots of opportunities for making mistakes, it is a good idea to include this information at least until it has been thoroughly checked. Form:

“incax”  
or  
“incax yes” include the a-x info;  
“incax no” a-x info not included

Default: info included

**incnler** In phase diagram calculations, there is the option of the set of reactions between the end-members of the minerals used in setting up the set of non-linear equations (equilibrium relationships) being included in the logfile. This script allows it to happen. Usually there is not much point in including this information. Form:

“incnler”  
or  
“incnler yes” include the reactions;  
“incnler no” reactions not included

Default: reactions not included

**Isqds** this script allows activities of end-members rather than mineral comps to be output for calculations. Form:

“Isqds”  
or  
“Isqds yes” end member activities output  
“Isqds no” mineral compositions output

Default: “Isqds no”

the output will look like this:

```
-----  
P(kbar)      T(?C)      kspL      anL      sill      foL      faL      h2oL      phl  
  8.80      866.2      0.169  0.00327  0.0398  1.18e-5  0.00215  0.250  0.0659  
  
mode      liq      bi      g      ksp      ilm      sill      q  
          0.3607      0  0.2377  0.09587  0.02994  0.02508  0.2508
```

**smath** If “makeo yes” is in the preferences file, indicating that a dr file is to be made (whose default name is tc-<name>-dr), this causes coordinate information from the calculations to be put in it. The default form of the coordinate information is Mathematica-readable (ie info being comma-separated, structured with curly brackets). This is the standard information used with the Mathematica “drawline” code for generating various diagrams. This can be varied, or changed via, “xinsmath yes” or “tabsmath yes”. Form:

“smath”  
or  
“smath yes” put coordinate info in dr file;  
“smath no” nothing put

Default: "smath no"

**xinmath** This qualifies "smath yes". Generally the coordinates output are just the PT. For some calculations, eg for T-x(CO<sub>2</sub>) diagrams, composition info is also needed. This script causes it to be output in addition to the PT info. Form:

"xinmath"  
or  
"xinmath yes" put extra coordinate info in "other" file;  
"xinmath no" nothing put

Default: "xinmath no"

**tabsmath** This qualifies "smath yes". Rather than the coordinate information being in Mathematica-readable format, it is output in simple tab-separated form (as understood by various spreadsheet programs). Form:

"tabsmath"  
or  
"tabsmath yes" do tab-separated output;  
"tabsmath ask" ask whether to produce tab-separated output;  
"tabsmath no" output not tab-separated

Default: "tabsmath no"

**cmath** If "makeo yes" is in the preferences file, indicating that a 'other' file is to be made (whose default name root is 'th dr' - Win: 'thdr.txt'), this causes compatibility diagram coordinate information from the calculations to be put in it. The form of the coordinate information is Mathematica-readable (ie info being comma-separated, structured with curly brackets), and is intended for use with the Mathematica functions for drawing compatibility diagrams. Form:

"cmath"  
or  
"cmath yes" put coordinate info in "other" file;  
"cmath no" nothing put

Default: "cmath no"

**txmath** If "makeo yes" is in the preferences file, indicating that a 'other' file is to be made (whose default name root is tc-<name>-dr), this causes Tx/Px pseudosection coordinate information from the calculations to be put in it. The form of the coordinate information is Mathematica-readable (ie info being comma-separated, structured with curly brackets), and is intended for use with the Mathematica functions for drawing these pseudosections. Form:

"txmath"  
or  
"txmath yes" put coordinate info in "other" file;  
"txmath no" nothing put

Default: "txmath no"

### **scripts for avPT (mode 2)**

**defsdparams** – for setting activity uncertainty defaults

**usedefsda** – using the defaults

**justrelu** - just relative uncertainties (as in the thermobarometry paper)

**setrockname** – allows the results to be labelled with the rock name

**diagnostics** – to always see the diagnostics

### **scripts for calculations not involving (solid) solutions (mode 3)**

**setmu** – allows you to set the chemical potentials of components

**seta / setlna / setloga** – allows you to set activities of components