The DataFetch Library of Functions for the Retrieval and Interpretation of Thermophysical Data from the TRC SOURCE Database[1]

R. C. Wilhoit[2]

The DataFetch library is a collection of functions which may be compiled and linked to user-written application programs. They extract information from a local version of the TRC SOURCE Database. The database is an extensive archive of experimental values of thermodynamic, thermochemical, and transport properties of pure compounds, chemical reactions, and binary and ternary mixtures. Application programs may call the library functions with parameters which specify the kind of data to be retrieved. The results are returned in a series of memory-resident buffers. The library functions operate at a series of processing levels. The lowest level returns the direct experimental values, along with associated information which identifies the components, properties, phases, literature references, sample descriptions, estimated uncertainties, etc. Higher levels return groups of related properties, normalized values of properties, selections of the most accurate values, and fits to smoothing functions. The library functions are written in C++ and interact with a local version of the SOURCE database. They are compatible with any platform which supports a C++ compiler in either a stand-alone or a client-server mode.

KEY WORDS: application programs; database; data interpretation; library functions; thermophysical properties.

1. INTRODUCTION

In the Thirteenth Symposium on Thermophysical Properties, we addressed barriers to the efficient utilization of thermophysical property data [1]. As the amount of experimental data accumulates in the world's literature, it

[1] Paper presented at the Fourteenth Symposium on Thermophysical Properties, June 25–30, 2000, Boulder, Colorado, U.S.A.

[2] Thermodynamics Research Center, Texas A&M University, College Station, Texas 77843, U.S.A. E-mail: rwilhoit@neo.tamu.edu

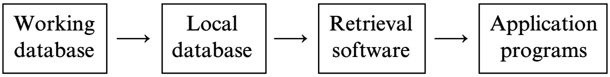
0195-928X/02/0100-0187/0 © 2002 Plenum Publishing Corporation

becomes increasingly more difficult and expensive to recover and use them. During the past two centuries this task has been eased by gathering data from scattered reports into compact compilations. These have taken the form of review articles published in journals, monographs, books, or sets of books. In recent years they often appear as computer readable databases. Some are issued as one-time publications and some as a continuing series of reports which attempt to keep up with current research. The kind of coverage, the method of presentation, and the extent of interpretation of the data vary widely among existing compilations.

Irrespective of the form of the compilation, those which reflect the state of knowledge of a subject at a particular time are static. To keep up with the world's accumulation of data, they must be periodically updated and reissued. The need to rework many of the same data for each revision and the difficulty in anticipating in advance what users may want causes wasted effort.

In our presentation 2 years ago, we described dynamic compilations. These are produced to order by the user at the time of need. They require two components. One is a continually updated computer readable archive of experimental data. The other is software which interacts with the archive, locates data of interest, interprets them, and converts them into a usable form. At that time, we described the SOURCE database, created and maintained by the Thermodynamics Research Center (TRC), which is suitable as an archive. The SOURCE database has since been improved and expanded in size.

In this concept information flows through four distinct components, as the following diagram illustrates.



The working version of the database is maintained in a central data collection facility and is continuously updated by various contributers. Information is downloaded periodically to compact read-only versions of the database located on individual workstations and servers. Retrieval software accesses the local database, extracts requested information, and passes it on to application programs. All user interaction takes place through the application programs. Through use of modern communication systems, it is also possible for the retrieval software to access the working version directly.

This model of information flow is valid for any large collection of data. Now we describe retrieval software specific to the TRC SOURCE database. It is in the form of a library of functions which we call the

DataFetch Library. These may be linked to application programs at compile time. Thus, application programs do not need to know about the details of database organization and access.

Application programs may then be written for many purposes. One kind simply displays data to the terminal screen, sends it to a disk file, or creates special-purpose databases. This type implements the dynamic compilation concept described in Ref. 1. Users can write their own application programs.

However, the model llustrated carries this concept further. Most users of thermophysical properties do not regard the numbers themselves as the final objective. Rather, they use these numbers to accomplish some further goal. Practical goals include the design and operation of manufacturing or transportation facilities, evaluation of new manufacturing processes, and improvement of health, safety, and environmental quality. Academic goals include studies of molecular structure–property relationships and testing of theories of matter. Programs written for such purposes can then obtain numerical data directly, without the need for human intervention.

2. GENERAL SPECIFICATIONS OF THE DATAFETCH LIBRARY FUNCTIONS

The DataFetch Library is a collection of computer functions which serves as the interface between application programs and the database. They are distributed in object form which may be linked to application programs during compilation. When an application program calls a DataFetch function, it passes parameters which specify the data to be retrieved. Results are returned in memory-resident buffers which may be accessed by the application program. The library can exist in either a static or a dynamically linked form. Application programs can be written to recognize the buffers, or front ends can be written for existing application programs.

The SOURCE database contains primarily directly measured values of properties of systems of specified composition. Sometimes smoothed or derived values are included, especially if the direct experimental values were not reported. It consists of 35 tables. Some store Registry Numbers[3] and compound names and formulas. Other tables contain literature references and author names. Tables for storing properties of pure compounds, binary mixtures, ternary mixtures, and thermochemical data and equilibrium constants for chemical reactions are present. Most kinds of thermodynamic and thermochemical properties of all phases and transport properties

[3] Numbers assigned by *Chemical Abstracts* are used whenever possible; otherwise numbers are assigned by the TRC.

of fluids are accommodated. Pure compounds and components of mixtures and reactions are identified by Registry Numbers. Data tables include values of state variables, properties, and estimated uncertainties. One table contains descriptions of samples used in the measurements. The database also includes a variety of metadata which describe the method of presentation, units in the original document, and other information to help in the evaluation and selection of data. The tables are indexed and linked so that related data may be retrieved. Properties and phases are identified by formally assigned codes. Detailed documentation of the SOURCE database is available [2]. Reference 3 gives an overview.

DataFetch functions which return data operate at a series of processing levels.

Level 1: Returns data for a requested property and system (pure, mixture, or chemical reaction) from a particular database table. These contain information directly extracted from the database with little change.

Level 2: Returns values of closely related groups of properties extracted from one or more relevant database tables.

Level 3: Returns data for a property group in a normalized form.

Level 4: Returns selected values. Normalized data from Level 3, which can now be intercompared, are screened to identify the best (most accurate) values.

Level 5: Returns parameters of smoothing functions fit to the output of Level 4 for a group of properties by the least-squares criteria, along with statistical measures of goodness of fit.

Level 6: Returns parameters of functions for calculating internally consistent properties. These satisfy thermodynamic constraints among properties. It operates on combinations of several Level 4 results.

Level 7: Carries out the same processing as Levels 5 and 6 but accepts a combination of Level 4 output with data from theoretical calculations or empirical correlations.

Application programs can call DataFetch functions at any level. Internally, functions at each level except the first use results returned from lower levels. Functions at Levels 1–4 return the requested numerical data, as well as associated information such as identification of properties and phases, sample descriptions, literature references, and available metadata. They closely reflect the organization of data in the SOURCE database.

3. ORGANIZATION OF THE SOURCE DATABASE

Database tables which store numerical values of thermophysical properties are defined by the number of components in the system and the number of associated independent state variables. The total number of state variables equals the degree of freedom of the system as calculated by the Gibbs phase rule, $F = C - P + 2$. C is the number of independent components in the system, and P is the number of phases. The database tables are based on the "effective" degree of freedom, which may be less than that calculated by the Gibbs phase rule. For example, if one or more state variables are kept constant for a set of data, they are not counted in the "effective" degree of freedom. The effective degree of freedom is also reduced by one for certain special states such as liquid–gas or liquid–liquid critical states and azeotropic states. Although the Gibbs phase rule does not apply to transport properties (viscosity, thermal conductivity, diffusivity), the concept of effective degrees of freedom is applied to these properties as well. Data with zero effective degrees of freedom are stored in tables which include all the descriptive metadata as well as the property values and uncertainties.

Data with more than zero effective degrees of freedom are stored in a pair of database tables. A header table contains information which describes the data. These include codes for the property and the state variables, the identification of phases, the sample numbers, and other metadata which describe the way the data were presented in the original document and which help in the interpretation. The value of any state variable which is kept constant for a data set is also stored in the header table. The numerical values of state variables not kept constant and the property are stored in a separate data table along with an uncertainty value for each property.

Both kinds of tables are indexed by registry numbers of components, a key for the literature reference, the property code, and a set number. A header table and its associated data table store the value of only one kind of property.

Calorimetric heats of chemical reaction are treated as having zero degrees of freedom and are stored in one table. Most equilibrium constant data are considered as having one degree of freedom where temperature is usually the state variable. They are stored in a header–data pair of tables. These thermochemical property tables are indexed by a reaction classification code and by registry numbers of four reaction participants. Registry numbers for any additional reaction participants as well as coefficients in the balanced chemical equations are stored in the header tables.

In principle, the distinction between a property and the state variables is arbitrary. For example, the pressure–temperature pair which defines an equilibrium between the liquid and the vapor phases of a pure component may be called a boiling point, where the temperature is the property and the pressure is the state variable, or a vapor pressure, where pressure is the property and temperature the variable. Another example is *P–V–T* data for a single-phase system of a pure component. Either pressure (P), volume (V), or temperature (T) may be chosen as the property, and the other two would be the state variables. In a set of isothermal *P–V–T* data, temperature is kept constant. The value of the constant temperature is placed in the header record, and the data values have one effective degree of freedom. In a set of isobaric *P–V–T* data, the pressure is constant, and in isochoric *P–V–T* data, the volume is constant. If nothing is kept constant, the property has two degrees of freedom. Sets of vapor–liquid equilibrium data in a binary system may be characterized by values of pressure, temperature, and composition of liquid and vapor phases (p, t, x, y data). The Gibbs phase rule gives two degrees of freedom for this system. Any one of these values may be considered as the property, and any two of the remaining three as the state variables. A complete description of the system requires two data sets for two choices of property. Altogether there are 24 ways of storing binary VLE data. If one of these variables is kept constant for a set of data, it then has one effective degree of freedom. Sets of data in which only pressure, temperature, and liquid composition are measured (p, t, x data) have only one property. A ternary system with vapor and liquid phases has three degrees of freedom. It is characterized by P, T, and two composition variables for each phase. Any of these may be chosen as the property, and any three of the others as state variables. Three data sets are required for a complete description. Altogether there are 180 ways of storing ternary VLE data.

Remarks in the preceding paragraph illustrate the fact that data may be stored in the SOURCE database in many ways. Any way that correctly describes the data may be used. Choices which reflect the way data are presented by the investigator in the original document are preferred. In any case, properties, state variables and phases are clearly identified in the header records for each data set. It is the responsibility of the retrieval software to sift through the database records, identify those relevant to a request, extract them, and present them in a uniform manner.

4. FURTHER DESCRIPTION OF PROCESSING LEVELS

Level 1 DataFetch retrievals extract data as they exist in the database. Classes in the C++ language have been defined for each of the following types of data: single-value, fixed-condition, one-variable, and two-variable data for pure compounds; single-value, one-variable, two-variable, and three-variable data for binary systems; single-value, one-variable, two-variable, and three-variable data for ternary systems; thermochemical data; and equilibrium constant data. Each of these has a virtual function called getd which extracts data of that type. Classes for retrieving compound names, literature references, and sample descriptions are also included in the library.

The use of these functions is illustrated by class pronev, for extracting one-variable data for pure compounds. The prototype for the function getd of this class is

```
void pronev::getd(unsigned long rgn, char *prc, char *phc, HDRAR&
  hdrar, DTAR& dtar, RFSAR& rfsar, REFAR& refar, SMPAR& smpar);
```

Parameters rgn, prc, and phc are passed to the function. Results are returned in the remaining parameters, which are references to container objects defined in the Standard Template Library of C++. These parameters are as follows.

rgn	Registry Number of compound
prc	Code for property to be retrieved
phc	Code for phase or phases
hrdar	Vector of structures for descriptive information
dtar	Vector of structures for numerical property data
rfsar	Vector of descriptions of reference states, if needed
refar	List of key values for literature references
smpar	List of key values for sample descriptions

The command,

```
pronev ovro
```

in an application program creates an object of type pronev and opens the related tables in the database. Subsequent commands such as

```
ovrc.getd(rgn,...)
```

or its equivalent using pointers will search the database and return the results in containers hdrar, etc. Subsequent calls to Level 1 functions cause results to be accumulated in the container objects.

Level 2 functions return data in a manner similar to Level 1. However, instead of passing parameters for specific properties and phases, parameters for various related groups of properties are passed. Results which may span more than one property and more than one table are returned in the container objects. Thus, for example, the *P–T* group of liquid–vapor phases of pure compounds would return data labeled both as boiling point and as vapor pressure. These data may reside in three different table combinations in the database. Similarly the *P–V–T* group for single phases of pure compounds combines all data independent of which variable was chosen as the property. The vapor–liquid group for binary systems would return all data involving pressure–temperature–composition variables for binary mixtures. Parameters for Level 2 functions include registry numbers and a code for the property group.

Level 3 normalizes the data returned from a Level 2 retrieval. Normalization makes a standard choice of property and state variables and converts them to a consistent form. The uncertainties listed for observed properties in the database are also propagated to the normalized form. The precise meaning of normalization depends somewhat on the property group processed by Level 2. Most properties which have zero degrees of freedom do not require normalization. Examples are critical temperature and pressure, triple-point temperatures, and normal boiling points of pure compounds. The vapor pressure–boiling point group for pure components are presented, with pressure as the property and temperature as the state variable.

Normalization of single-phase *P–V–T* data for a pure compound would rearrange the results returned from Level 2 so that the property was a particular volumetric property such as density, and temperature and pressure, in that order, were the state variables. All uncertainties for whatever properties were selected in the database would be converted to the uncertainty in density.

Normalization of data for mixtures is more complicated. For example, volumetric properties of mixtures, in addition to those listed above, include excess volume, partial molal volume, and apparent molal volume. Compositions of mixtures may be expressed as mole fraction, mass fraction, volume fraction, molality, molarity, etc. Normalization converts all of these into one kind of composition variable.

Normalization is a critical step which is required for all higher levels of data handling. Normalization may require complicated conversions and reorganization. It may also require auxiliary data, not part of the property group being normalized. The auxiliary data may be obtained from a separate Level 5 retrieval for the auxiliary properties or from the parameter database discussed below. Propagation of uncertainties into the final choice of property requires a knowledge of derivatives of the property with respect to state variables. It may be necessary to carry out a Level 5 calculation using some nominal uncertainty estimates to obtain the derivatives. This sets up a loop in which the propagation of uncertainties, the fit to a smoothing function, and the calculation of derivatives are repeated. It should converge after a few iterations.

Level 4 operates on data returned from Level 3. It selects the best (most accurate) values of properties whenever duplications, or near-duplications, exist among the reported properties. This screening is based primarily on the estimated uncertainties of properties, either given directly in the database or propagated to the normalization data.

The selection of data with zero effective degrees of freedom is simple. It is only necessary to establish an upper limit for the uncertainty and to select data whose uncertainty is less than the limit. Selection of data which have one or more degrees of freedom is more complicated. It is necessary to consider not only the uncertainty limit but also the way the data are distributed over state variable space. An effective algorithm has been developed by the TRC and has been used for several data evaluations. It is described in several published references [1, 2, 4], and has also been used in an extensive review of virial coefficients now in press.

Briefly, the selection of each property value at a particular point in the state variable space is based on the comparison of the uncertainty assigned to that point with a weighted mean of uncertainties assigned to neighboring values in the state variable space. The weighting factor is an inverse exponential function of the difference between the state variables of the property in question and the state variables of each neighboring point. The selection level and the parameters in the weighting function depend on the size of the data set, the range of state variables it covers, the kind of property being selected, and other considerations.

Data with one or more degrees of freedom returned from Level 4 are now suitable for fitting to smoothing functions or theoretical models by the least-squares criteria. Level 5 returns parameters for the selected model as well as statistical measures of the goodness of fit. Reciprocals of the square of the uncertainties in properties serve as weighting factors for the fit. Combination of the parameters with Level 4 data permits comparison of the selected property values with those predicted by the model.

Level 5 fits functions to the group of properties defined in Level 2 and returned from Level 4. This automatically guarantees internal consistency among these properties. However, these results will not necessarily be thermodynamically consistent with other kinds of properties. To achieve a global internal consistency, it is necessary to fit all related properties to a general P–V–T or Helmholtz energy equation of state. Level 6 carries out multiproperty fits of this kind by combining several sets of data from Level 4 retrievals.

The extent of experimental data may not be sufficient to support satisfactory operation of Level 5 or Level 6 functions. It may be then possible to supplement Level 4 results with data calculated by theoretical or empirical correlation techniques. The combined data sets can then be passed to Levels 5 and 6 functions. These are considered Level 7 results. An example would be the combination of ideal-gas thermodynamic functions calculated by partition functions based on molecular energy states by spectroscopy [5] with experimentally derived ideal gas heat capacity and entropy from the Level 4 function. Calculated ideal-gas functions could also be included in the Level 6 step. Group additivity correlations are available for ideal-gas functions [6], critical constants [7], and enthalpies of formation [8], among others.

5. IMPLEMENTATION OF THE DATAFETCH LIBRARY

At the present time Level 1 DataFetch functions have been written and tested for all properties They access a local version of the SOURCE database created by the c-tree Plus®file management functions distributed by the FairCom Corporation. It supports single- or multiuser access and client/server operations on a wide range of platforms.

A package consisting of the database, six application programs which run under the Linux operating system, and supporting documentation is available from the TRC [2]. One program searches compound names and formulas to find registry numbers. The others retrieve data for pure compounds, binary mixtures, ternary mixtures, thermochemistry, and equilibrium constants. Formatted output is sent to the terminal screen or disk file. The results are grouped into four sections: Property Descriptions and Metadata, Data Values including uncertainties, Sample Descriptions, and Literature References. Users can also write application programs to be linked to the DataFetch library.

Work is in progress on Level 2 functions. Versions for other UNIX and for the Windows operating systems are planned.

REFERENCES

1. R. C. Wilhoit and K. N. Marsh, *Int. J. Thermophys.* **10**:247 (1999).
2. *Documentation for the TRC Source Database* (Thermodynamics Research Center, National Institutes of Standards and Technology, Mail Stop 838.00, Boulder, CO 80305-3328, Feb. 20, 2001). It can be downloaded from the TRC web site, http://trc.nist.gov.
3. M. Frenkel, Q. Dong, R. C. Wilhoit, and K. Hall, *Int. J. Thermophys.* **22**:215 (2001).
4. R. C. Wilhoit, K. N. Marsh, X. Hong, N. Gadala, and M. Frenkel, *Landolt-Börnstein, Group IV. Physical Chemistry, Vol. 8. Thermodynamic Properties of Organic Compounds and Their Mixtures, Subvolume B. Densities of Aliphatic Hydrocarbons. Alkanes* (Springer-Verlag, Berlin, 1996). Also *Subvolumes C–F.*
5. M. Frenkel, G. J. Kabo, K. N. Marsh, G. N. Roganov, and R. C. Wilhoit, *Thermodynamics of Organic Compounds in the Gas State, Vols. I and II*, TRC Data Series, (CRC Press, Boca Raton, FL, 1994).

6. S. W. Benson, F. R., Cruickshank, D. M. Golden, H. E. Haugen, H. E. O'Neal, A. S. Rodgers, R. Shaw, and R. Walsh, *Chem. Rev.* **69**:279 (1969); S. W. Benson, *Thermochemical Kinetics*, 2nd ed. (John Wiley, New York, 1976).
7. A. L. Lyderson, *Engineering Experiment Station Report No. 3* (University of Wisconsin, Madison, 1955); K. G. Joback and R. C. Reid, *Chem. Eng. Commun.* **57**:233 (1987); G. R. Somayajulu, *J. Chem. Eng. Data* **34**:106 (1989).
8. J. B. Pedley, *Thermochemical Data and Structures of Organic Compounds, Vol. 1,* TRC Data Series (CRC Press, Boca Raton, FL, 1994).