```
HHHH    HHHH   TTTTTTTTTTTTTT   BBBBBBBBBBB              22222
HHHH    HHHH   TTTTTTTTTTTTTT   BBBB     BBBB         22222 22222
HHHH    HHHH        TTTT        BBBB      BBBB     2222        2222
HHHH    HHHH        TTTT        BBBB      BBBB      222         2222
HHHHHHHHHHHH        TTTT        BBBB     BBBB                6222222
HHHHHHHHHHHH        TTTT        BBBBBBBBBB               2222222
HHHHHHHHHHHH        TTTT        BBBB     BBBB          22222
HHHH    HHHH        TTTT        BBBB      BBBB       22222
HHHH    HHHH        TTTT        BBBB      BBBB      2222         22
HHHH    HHHH        TTTT        BBBB     BBBB      2222222222222222
HHHH    HHHH        TTTT        BBBBBBBBBBB        2222222222222222
```

HTB2 - A MODEL FOR THE THERMAL ENVIRONMENT OF BUILDINGS IN OPERATION

TECHNICAL REFERENCE MANUAL

release 1 rev 0   apr-85

VOLUME 1

D.K.ALEXANDER & Dr P.T.LEWIS
Welsh School of Architecture, UWIST, Cardiff

HTB2 - A MODEL FOR THE THERMAL ENVIRONMENT OF BUILDINGS IN OPERATION

TECHNICAL REFERENCE MANUAL

This manual is presented in three sections, within two volumes;

Section 1 describes the structure and conceptual basis of HTB2 and introduces the implementation of these concepts.

Section 2 provides a complete description of each routine used in the current implementation, organised in the conceptual modules introduced in section 1.

Section 3 provides a description of the database as used by these routines.

Appendices describe the installation, I/O usage and output formats of HTB2.

Volume 1 contains section 1 and section 2 (chapters 1 to 10).
Volume 2 contains section 2 (chapters 11 to 17), section 3 and the appendices.

This document is not intended as an user manual, but as a reference manual for the description and implementation of the concepts used in the model.

This document refers to HTB2 release 1 (apr 85) revision 0.

SECTION 1

GENERAL INTRODUCTION TO HTB2 - CONCEPTS AND IMPLEMENTATION

This section will provide an introduction to the concepts and implementation of the building model HTB2. It should be read before the more detailed technical description presented in section 2.

CHAPTER 1 INTRODUCTION

1.1 THE NEED FOR NEW BUILDING MODELS 1-1
1.2 HTB2 AS A RESPONSE 1-2
1.3 SUMMARY OF FEATURES AND ASPECTS OF HTB2 1-3

CHAPTER 2 HTB2 STRUCTURE

2.1 STRUCTURE - PARTITIONING OF TIME 2-1
2.2 STRUCTURE - PARTITIONING OF PROCESSES 2-3
2.2.1 Initial Conditions - Model Database 2-3
2.2.2 Surface Radiant Transfer 2-4
2.2.3 Meteorological Database 2-4
2.2.4 Insolation 2-4
2.2.5 Fabric Heat Transfer 2-5
2.2.6 Ventilation Transfer 2-5
2.2.7 Response Of Controls And Heating Systems . . . 2-5
2.2.8 Incidental Gains 2-6
2.2.9 Operation Scheduling 2-6
2.2.10 Data Output 2-6

CHAPTER 3 HTB IMPLEMENTATION

3.1 PROCESSES 3-1
3.1.1 Initial Conditions - Database 3-1
3.1.2 Surface Radiant Transfer 3-3
3.1.3 Meteorological Database 3-4
3.1.4 Insolation 3-4
3.1.5 Fabric Heat Transfer 3-5
3.1.6 Ventilation Transfer 3-6
3.1.7 Response Of Controls And Heating Systems . . . 3-7
3.1.8 Incidental Gains 3-8
3.1.9 Water Vapour Generation And Transport 3-8
3.1.10 User Routine Link Points 3-9
3.2 INPUT AND OUTPUT 3-9
3.2.1 Input Data Organisation 3-9
3.2.2 Diary Scheduling Input 3-10
3.2.3 Output Procedures 3-10
3.2.3.1 Block output reporting 3-11
3.2.3.2 Event logging 3-11
3.3 HTB2 EXTRA FEATURES 3-12
3.3.1 Virtual Spaces 3-12
3.3.2 Virtual Partitions 3-12
3.3.3 Insolation Control 3-12
3.3.4 Diary Control 3-13
3.3.5 Geometric Considerations 3-13
3.3.6 User Link Points 3-13
3.3.7 Routine Identification 3-13
3.4 HTB2 PORTABILITY 3-14
3.4.1 Use of Include Statement 3-14
3.4.2 Machine Dependant Routines 3-14

CHAPTER 1

INTRODUCTION

1.1 The Need For New Building Models

HTB2 has been designed as an investigative model of the thermal performance of buildings. It has been developed in the light of many years research, in both modelling and monitoring, on the thermal environment of modern low energy buildings. Through much of this work it has become apparent that, in practice, the influence of fabric on the gross energy demand of buildings has decreased as higher insulation levels and stricter controls over infiltration have been implemented. Due to these changes the characteristics and performance of ventilation and heating systems and their controls, the utilisation or control of solar energy and incidental gains, and the interaction of occupants with the building system, have all become increasingly important in the total performance of the building.

A general reduction in the gross energy requirement of buildings has thus been achieved, partly through the application of results from past models, and will continue. It is felt that the further application of building models in research will require a shift in concentration from performance as gross energy usage to performance of the built environment and the operation of the building as an interacting system. As this system is comprised, in part, of climate, fabric, ventilation, controls, plant, incidental heat sources, and occupants, an ideal simulator of the built environment will obviously be required to model each of these sub-systems to a equal level.

Further, a shift in time-scales must also take place. In the types of application considered for such a research model, for instance in the investigation of heating system controls and building constructions, detail at a time scale of minutes will be at least as important as seasonal values.

As the modelling of many of the sub-systems mentioned above are in an early stage of development, any model produced with this type of application in mind must be easily upgraded and enhanced as further techniques and information become available. This adaptability must be seen as a product of high programming and documentation standards as well as an inherently flexible implementation design.

1.2 HTB2 As A Response

In the past many building thermal models have concentrated on the fabric transport of heat, and were designed to produce yearly or seasonal energy requirements. HTB, the precursor of the present model, was such a model. It provided data on the hourly, daily, and seasonal temperatures and heating requirements of buildings, based in the main on dynamic fabric heat loss. Ventilation, occupancy, and incidental gains were considered, but only on a simplified scale.

HTB2 was developed to model the detailed workings of a building, incorporating the many aspects of thermal transport and gain, providing data on short time-scales of minutes and weeks, rather than hours and years. It has been built around the explicit finite-difference fabric transport model of HTB, but has been extensively enhanced in the determination of solar and incidental gains, ventilation transport, heating system and control responses, and the specification of operation schedules (i.e. occupant intervention) at short time-scales. Data output has been improved to allow significant events, for example a change in air temperature, to be recorded at a short time-scale without a massive increase in output quantity due to redundant information.

As indicated HTB2 is intended primarily as an investigative research tool, rather than a simple design model. It is envisaged in use in the development of the understanding of the operation of the building system and as a test bed for future model development. Although capable of the functions required of simpler design oriented models, such as seasonal energy requirements, the general nature and organisation of HTB2 does make this type of application less attractive than other approaches, should that particular result be all that is required. It has been found however that the extra information and detail available from the more complex models is often found to be of interest and use, once that information is made available. It is anticipated that the main user of HTB2 will be the scientist/programmer, perhaps working in conjunction with a research architect or design team.

The structure and implementation of HTB2 reflect this type of application. Understanding of the implementation, alterations, and improvements will hopefully all be achieved more easily than was the case with past models. HTB2 is written in standard Fortran-77 to improve portability, and to improve the structure and standard of the programming. It is also written in a modular format in which, as far as feasible, each sub-system has been isolated and localised in specific subroutines so that areas of interest may be quickly identified. Earlier models such as HTB have tended to be written as seamless programmes, each process interwoven, restricting the ability to make even simple alterations or improvements without intensive, or time consuming, effort. As a completely generalised building model is difficult and perhaps impossible to produce, each use on a new research problem generally requires some development effort to be made before productive work can be done.

The programming of HTB2 has been made as explicit as possible to enable the easier understanding of its internal operation. This has necessarily been achieved at the expense of efficiency of computation but it is felt that in a research environment computer time is of less importance than development time. For similar reasons data structures are not as compact as they conceivably could be, as memory size has also been traded off against ease of understanding. In many computing requirements this apparent loss of operational efficiency is not as serious as it would seem, as many modern FORTRAN compilers have optimizing capabilities often beyond those of FORTRAN programmers, and virtual memory systems ease memory restrictions.

1.3 Summary of features and aspects of HTB2

- portable coding
- designed for easy alteration and extension
- investigation of short time-scale phenomena
- external meteorological data
- convective and radiant gains and exchange
- internal fabric temperature and heat fluxes
- provision for imposed internal temperature patterns
- inter-space air movement
- water vapour gains and transport
- provision for zoning of spaces
- shading
- provision for placement of solar gains
- flexible glass transmission specification
- flexible specification of operating schedules
- flexible output intervals and details

CHAPTER 2

HTB2 STRUCTURE

This chapter outlines the conceptual basis and structure of the, perhaps idealistic, model HTBn. The current implementations of these concepts in the programme HTB2 are discussed in chapter 3.

2.1 Partitioning Of Time

The basic problem facing a model of the built environment is the reduction of the complex web of interactions which comprise the transport and gain of heat in the system, into a network of calculation procedures (fig 1a). HTBn achieves this by utilising the partitioning of time into discrete intervals inherent to the explicit finite difference fabric transport model at its core. Within each of these intervals, initial (or driving) conditions of temperatures and heat fluxes are held to remain constant and all transport mechanisms can therefore operate independently (fig 1b). New conditions for the next time interval are determined from the accumulated effects of these mechanisms, thus allowing interactions to be determined over a multiple of the basic time interval.

The actual time interval used depends on the accuracy required and on the stability criteria of the finite difference calculations used for fabric heat transfer. This stability depends on the materials and constructions to be modeled and is generally expressed as

$$0.5 \geq \frac{dT * k}{D * C * dX^{2}}$$

where dT is the time interval , sec.,
C is the material specific heat capacity , J/Kg/oC,
D is the material density , Kg/m3
k is the thermal conductivity , W/m2/oC ,
and dX is the inter-node thickness , m.

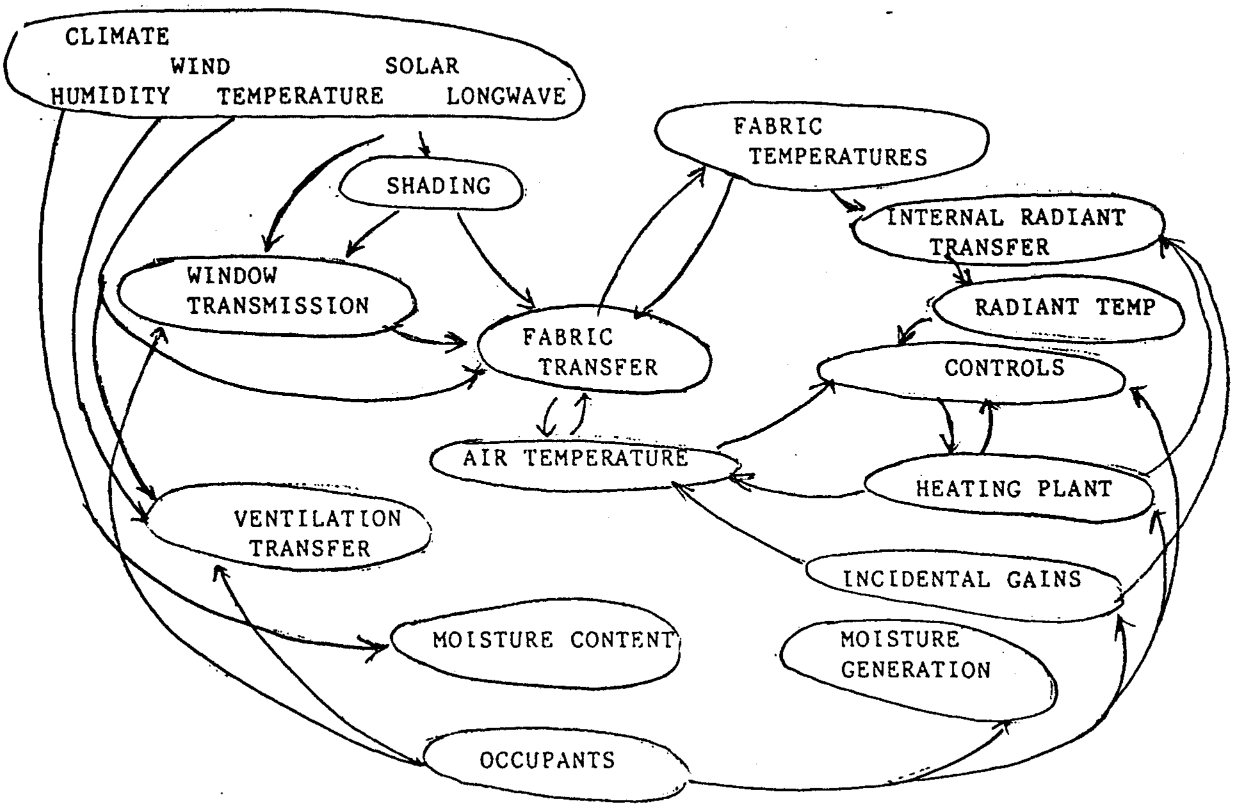


figure 1a : FUNDAMENTAL BUILDING PROCESSES AND INTERACTIONS

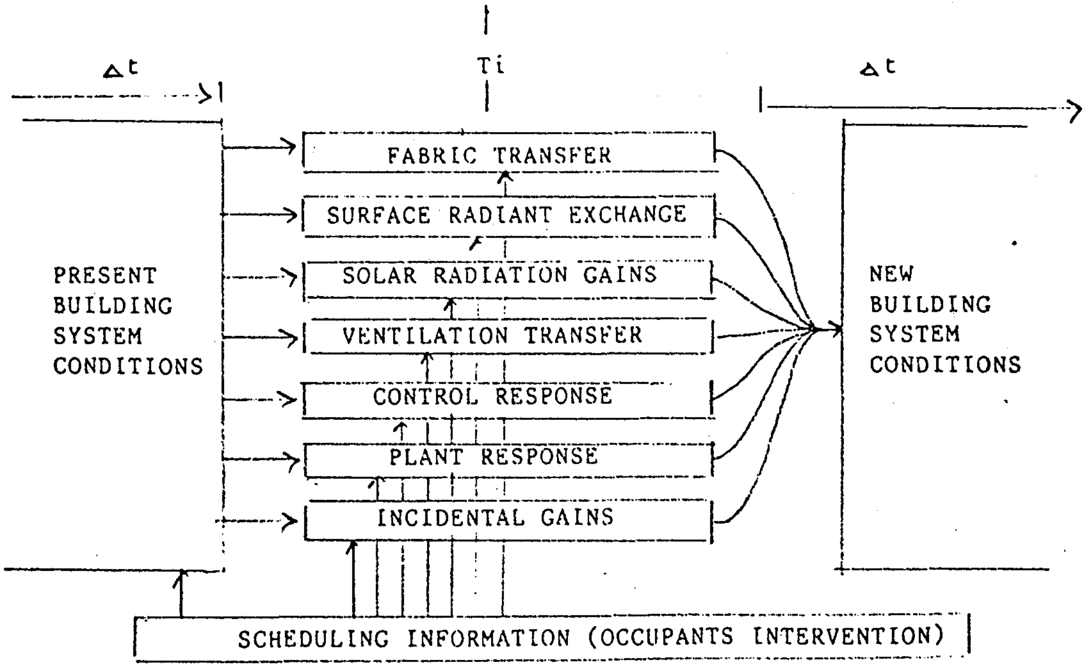


figure 1b : PARTITIONING OF TIME AND PROCESSES

Typically, time intervals of 10 - 60 seconds are expected. This must represent the smallest time-scale considered in any process. No effects below this time limit are modeled, so that calculated or initial values represent mean or steady results over the interval.

Many processes will have useful time-scales much longer than this fundamental interval. Therefore larger, encompassing, time intervals allow for the updating of meteorological information (typically hourly data is available from meteorological databases), calculation of ventilation airflows (on the order of 10 minutes, from consideration of the necessity to assume good mixing), and reporting of calculated data.

2.2 Partitioning Of Processes

As each process involved is treated as acting independently (within the current time interval) on the information produce by the previous time interval, the calculations are basically circular in operation. The calculations may be summarised by ;

- determine new space temperatures from last round results,
- determine radiative transfer between internal surfaces,
- determine external conditions,
- determine radiative fluxes due to insolation,
- calculate fabric heat transport,
- calculate ventilation heat and moisture transport,
- determine response of controls and heating systems,
- calculate incidental gains of heat and moisture,
- refer to building operation scheduling information,
- output data,

and repeat for length of simulation required.
The processes involved in these stages are described in greater detail below.

2.2.1 Initial Conditions - Model Database

Data maintained in the database include fabric temperatures (both internal and surface), space air temperatures, net convective heat gains to internal spaces, irradiance to surfaces from incidental sources, and direct heat gains to fabric, all as determined from the previous round of calculations (or as initially set at start-up). The net convective heat gains to spaces are used in conjunction with the heat capacity of the space air to determine new internal air temperatures. These factors become the basic internal driving forces for the transfer processes of the present time interval.

2.2.2 Surface Radiant Transfer

The transfer of energy at surfaces, both internal and external, due to irradiance from systems, incidental sources, insolation, and longwave exchange is determined here. The result is a net radiant gain of energy for each surface in the building system.

2.2.3 Meteorological Database

External driving forces (external air temperature, solar irradiances, etc.) are determined from supplied meteorological data.

At specified intervals new meterological information is read from an external data file. The meteorological database (derived from this information) is comprised of;

- air temperature
- ground temperature
- wind speed and direction
- humidity and vapour pressure
- sun position
- direct and diffuse horizontal irradiance
- direct normal irradiance
- convective transfer coefficients for external surfaces
- unshaded solar irradiance to external surfaces
- longwave irradiance to external surfaces

These parameters are held as constant over the meteorological time interval. This is normally one hour but any interval may be specified , governed by the availability of meteorological data.

2.2.4 Insolation

The known sun position and the unshaded (i.e. maximum) irradiances on each surface are used in conjunction with the specification of shading and material transmission, to determine the solar irradiance on external surfaces, the insolation through transparent constructions, the absorbed energy within a transparent element, and the solar irradiances onto internal surfaces. The result is a irradiance to each surface, and an absorption to each layer of each element.

2.2.5 Fabric Heat Transfer

For each element (i.e. a wall or ceiling) of the building system, surface conditions of air temperature and irradiance, and net direct energy gains, have by this time been established. Here, fabric energy transport due to these boundary conditions is calculated. According to the orientation and environment of the surfaces a convective heat transfer coefficient is selected and the heat flow between the layers of the element and the new layer and surface temperatures are determined. The convective heat transferred from internal space air is accumulated to be applied in the net convective heat gain to the space for the next time interval.

2.2.6 Ventilation Transfer

Ventilation transfer is considered to be due to airflows between internal spaces and between an internal space and the exterior. These ventilating airflows are used to determine the net exchange of heat (and moisture) for each space. As in the fabric process the net convective gain of a space is accumulated to be applied in the next time interval.

The usual assumption of good mixing in each space or zone is made in these calculations so that air movement or temperature differentials within a space are not considered. It is possible however to divide a large open space into smaller zones with imaginary partitions so that a temperature distribution may be simulated.

2.2.7 Response of Controls and Heating Systems

This stage determines the amount of heat output by a specified system in response to the current conditions and the past history of the system. The output of each system may be convective gain to specified spaces, irradiance to specified surfaces, and direct heat gain to specified layers of elements. The system output is determined in response to the current heating status, as determined by a time clock or operation scheduling, the output of an associated controls and sensors, the systems maximum capability at the time, and the response characteristics of the system to change.

2.2.8 Incidental Gains

Incidental gains to the building are divided into four categories; lighting, occupants (physiological), small power, and others. Each of these categories is specified and treated independently, and the result of each is a net convective heat gain to each space, a net irradiance to each surface, and also a net water gain to each space.

2.2.9 Operation Scheduling

Scheduling information controls the operation of several of the sub-systems of the building; the heating plant, incidental gains, ventilation, etc. To enable the simulation of buildings in use, the specification of these schedules must be flexible enough to allow realistic scenarios to be simulated. Changes must be able to be made at any specified time, relative to the time of day maintained within the simulation. Further, there must be control over much of the information held within the model; thermostat settings, time-clock controls, incidental gain sources, and even the building partitioning must be open to alteration during a simulation.

An scheduling structure appropriate for this requirement is the diary, that is a list of times and actions to be taken. In effect scheduling becomes event driven, where an event may be, for instance, heating system controls being switched on or off, or the alteration of a thermostat setting.

2.2.10 Data Output

The database of the model will contain, in addition to the more common items such as internal temperatures or heating loads, a large amount of information which will be of use or interest only in particular applications. These items may be, for instance, temperature and heat flux profiles through elements, or system duty cycles. The general output procedures of the model must cater for these items but at the same time be controllable so that unnecessary data is not produced. Also data must be available at the time-scale of interest, hourly output blocks are of little use if the warm-up characteristics of the building are being investigated, but the production of huge volumes of largely redundant data must be avoided.

Two output procedures are required; interval block reports of standard items, and an event driven log of selected items. The interval of the block report will be controllable, while the event logging procedure will produce output on the selected items only when their values change by a preselected amount. For instance, in the event log internal air temperature may be recorded only when its value changes by 0.5 degree C or more from the last report. As the time of the event is also included in the record, this in effects produces information on rate of change of the item from which the complete time history may be reconstructed.

CHAPTER 3

HTB2 IMPLEMENTATION

This chapter deals with implementation of the concepts outlined in the preceding one , introducing information specific to the current implementation of HTB2. A more detailed discussion of each area is presented in section 2, the description of the component subroutines of HTB2.

HTB2 has been developed from the model HTB and at present incorporates many of the earlier models simplifications. However the structure of HTB2 has been designed in such a way to allow easy incorporation of new and more accurate algorithms or procedures as they are developed.

3.1 Processes

The conceptual stages are treated, within any one interval, independently and so are isolated in separate managing subroutines. It is felt that as well as producing a more easily maintained model, this structure is also intuitively more easily understood, thereby hopefully reducing the learning time involved in operating a complex model.

An operational block diagram of HTB2 is given in figure 2. The overall managing routine described in section 2 is HTBOS.

3.1.1 Initial Conditions - Database

All the relevant information about the structure and specifications of the building, and of the current conditions (i.e. of temperatures, status etc) are contained in several common blocks comprising the model database. The various subroutines for the individual processes may thus easily access the data without the need for complex call arguments.

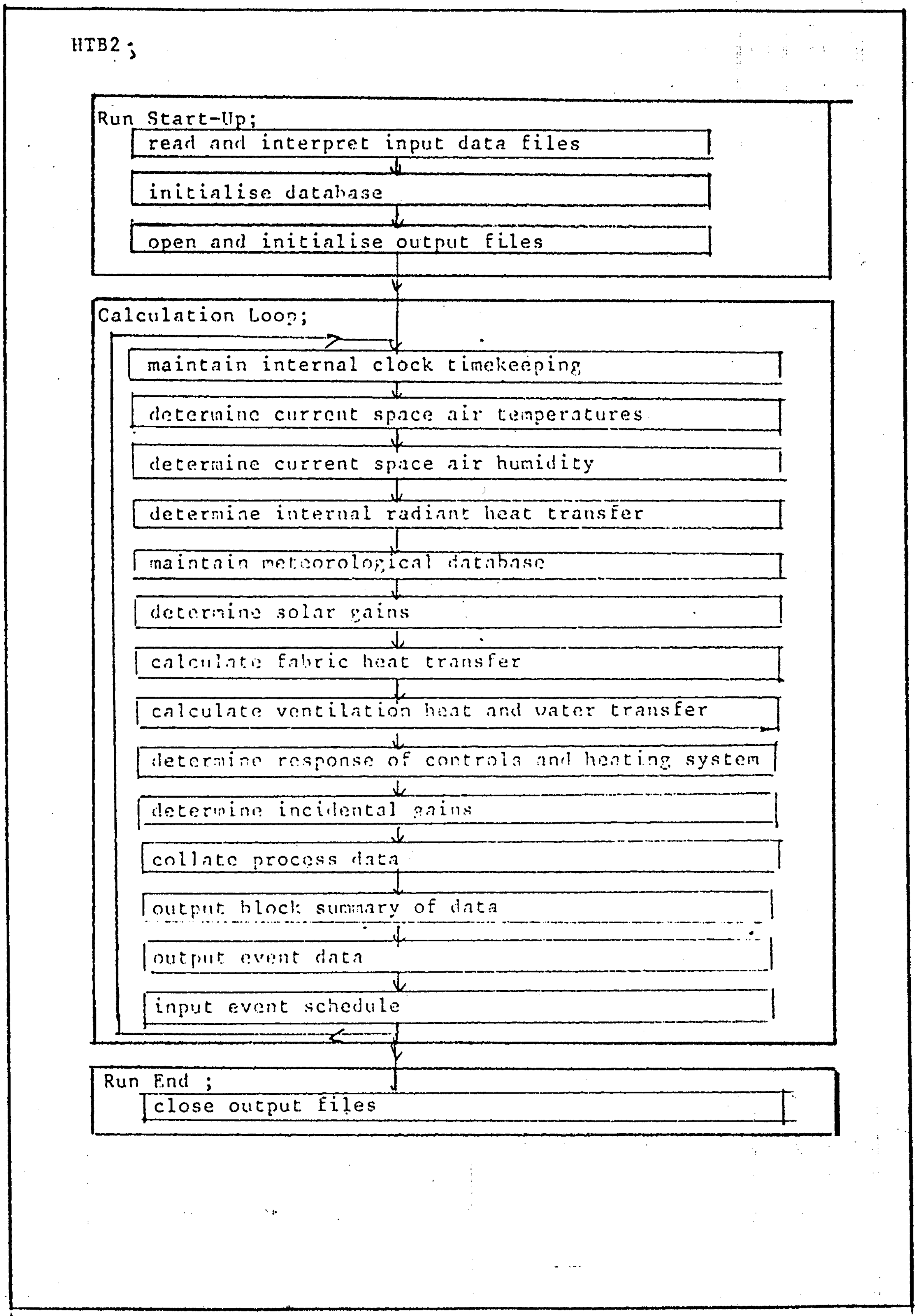


figure 2 : HTB2 BLOCK STRUCTURE

The fixed nature of data structures in Fortran necessarily imposes maximum data sizes for any one construction of the model, however these may be easily altered at compile time through the use of parameter statements and included parameter definition files in array sizing. There is, for instance, the capability of specifying the maximum number of spaces expected, or the maximum number of elements, heating systems, etc. Of course any data set less than these maximums may be specified.

A complete discussion of the database is given in section 3.

Calculations at this stage include the determination of new space air temperatures from the accumulated convective heat gains of the past round of calculations, and the management of internal clocks for the current time of day, in both seconds past midnight and a 24-hour clock, and the day of the week, date, etc. as related to the starting date specified at start-up.

Associated subroutines are ;
CLOCK , TEMPSP , UPDATE

3.1.2 Surface Radiant Transfer

A choice of two methods of calculation will be available, selected by an input option set at start-up. Both determine the interchange of longwave radiation between internal surfaces, and the net absorption of irradiance produced by incidental sources and insolation (from the values determined in the preceding time interval).

The first is an approximate solution determined, for the longwave surface exchange, by deriving a star circuit based on the area-emissivity weighted radiosity of each surface, and for radiant sources, an area-emissivity weighted assignment of the reflected component.

The second method is still in development but will provide a method of solution of the total radiant network using matrix methods and surface view-factors. The necessary linkage points for this method are included in the current implementation, but the routine will not allow itself to be called.

It is felt that the approximate solution will be sufficient for most purposes, however the extra accuracy of the second approach may be necessary in modelling modern lightweight surfaces.

Associated routines are,
TEMPRD , EXCTRD

3.1.3 Meteorological Database

Required external meteorological data, as input from a data file, are air temperature and humidity, ground temperature, wind speed and direction, total and diffuse horizontal irradiance, and cloud cover. This data is presented to HTB2 in a simple textual ascii file, and so may represent data determined from meteorological office databases or imaginary, artificially produced data for specific applications.

Calculated parameters such as sun position, direct normal and surface irradiance are determined from standard textbook algorithms.

External surface heat transfer coefficients are selected from handbook values according to the exposure of the surface and stored for later reference. The form of access to the stored coefficients is in a function call so that a method of direct calculation may be easily introduced.

Similarly the external longwave radiant transfer at a surface is also pre-calculated and stored for later access by a function call.

Associated routines are,
METBAS , EXTRHT , LNGWAV

3.1.4 Insolation

The shading of external surfaces is determined from a pre-defined masking template for each external element. This template is a 10 degree sector matrix of the view of the sky vault as seen by the surface, the matrix containing the attenuation of the unshaded irradiance from that sector. The production of this matrix is external to HTB2. It could, for instance, be produced from a stereographic projection of the site layout.

Windows are not differentiated from opaque elements except in the specification of transmission coefficients, thus each window may also have a shading template. The transmission specification describes the transmission and absorption coefficients of the composite structure of the element for angles of incidence in 10 degree steps. These specifications may be obtained from manufactures published data, or may be theoretically derived. As in the case of the shading template, the method of production for these values has been kept independent from HTB2 to enhance its flexibility.

The transmitted solar energy finally determined may be apportioned to specified elements to produce the solar irradiance at each internal surface. At present this apportioning is simply specified by list, however a time-dependant sun-patching routine taking sun position into account may be incorporated at a later date.

These calculations are normally carried out only when new meteorological data is presented. That implies the solar influx is held to be constant over the meteorological interval. There is provision, however to demand recalculation at any time, for instance if the shading template is altered to simulate alteration of blinds. This is controlled by logical flag, and it is the responsibility of the routine which makes the alterations to request the recalculations.

Associated routines are,
SOLAR , SKYVW , TRFNCT

3.1.5 Fabric Heat Transfer

Heat transfer through the fabric is calculated via an explicit finite difference method. The building fabric description contains information about each separate material layer within each element. This approach allows the calculation of the heat flux and temperature profiles though each element, which is then available for output. This approach may also put severe limits on the time-step needed for calculation stability, however as already mentioned it is envisage that the model will be used for short time-scale investigations so that this limitation is not felt to be critical. As an operational aid HTB2 will determine the optimum node partitioning required for the time-step specified and will abort its run if stability cannot be achieved.

Calculations are made on the level of the element, so spatial (i.e. 3-dimensional) information is not required. In the terminology of HTB2 an element may be a wall or part of a wall, or a ceiling, floor, window etc. , and can connect at most two spaces. A space is any air volume which can, or will, be considered as a uniform air temperature. The relation between modeled spaces and the actual rooms of the building is flexible and perhaps most dependant on the level of information required from the simulation; a large room could be zoned into smaller spaces while a number of small rooms could be engulfed in a larger zone. The external air is considered as a space so that an element may connect a space with another space, with the exterior, or even with the same space

(the latter case would represent a totally internal mass, acting as a thermal store).

The finite-difference procedure used provides data on the net heat transferred from the connecting spaces, the surface temperatures, and the internal temperatures and heat fluxes within the materials of the element.

Each element is described by its orientation, the spaces it connects, and its structure. The stucture is a description of the material properties (thermal conductivity and capacity) and thicknesses used in the construction of the element. Further information on each element include the surface properites of reflectance, transmittance, and emissivity. There is a special class of elements (called 'virtual') which are imaginary partitions and transfer heat directly from air on one side to air on the other. Such a partition takes no part in radiant transfer calculations.

Surface convective heat transfer coefficients are determined by reference to function calls according to the type of surface; internal, external or 'virtual'. The values returned are currently handbook values chosen according to surface orientation.

Cavities in constructions are not treated as a part of the finite difference procedure. Instead they are currently treated as a thermal resistance between the two surfaces. They are currently treated as unventilated, but an extension to allow air communication with one space (or the exterior) is being tested.

Ground floors of buildings may be exposed to external air temperature or alternately to a deep ground temperature as supplied by the meteorological data. In cases where ground temperature is not available, HTB2 will calculate an effective ground temperature from a time-constant applied to the external air temperature.

Associated routines are,
FABRIC , HTCF , HTCE , HTCX , CAVTRN

3.1.6 Ventilation Transfer

Ventilating airflows may be determined or described in several ways, depending on the data available and the information required. Each methods exists as an option within HTB2, which is selected at run-time.

Ventilation may for instance be specified as a simple air-change rate (i.e. an air exchange between the space and the exterior only) for each space, or as a network of space to space (and space to exterior) flow-rates. For these options, up to three different rates or patterns may be specified, the choice being made in the operating schedule.

Provision is made to enable empirical air-change rates to be specified (as a function of wind direction, speed and square root of temperature difference) and a mathematical ventilation multi-cell model is will be included (at time of writing this is still being tested).

Specified airflows are checked each time-step for alterations, while airflow models are only recalculated at specified intervals, or on demand according to changing conditions.

Associated routines are,
VENTL , VENT1 , VENT2 , VENT3 , VENT8

3.1.7 Response of Controls and Heating Systems

Each heating system, of which there may be a number, is described by

- maximum output at the present time,
- warm-up and cool-down characteristics,
- time delay to changes,
- output connections, and
- control characteristics which are in turn described by
  - control criteria,
  - sensor response, and
  - sensor location.

The output of each heating system may be connected to a variety of spaces (for convective output), surfaces(radiant), or element layers (direct). A system may be specified as having a time-lag to changes in control output, and separate exponential warmup and cool-down characteristics. Controls include a time-clock and manual override, and a thermostat the sensor of which may also have specified time-lags and exponential characteristics, as well as a deadband and an accelerator heater. The sensor may have specified coupling characteristics to air, radiant and surface temperatures, and may produce output proportional to the shortfall in temperature or to a temperature band, or a simple on/off output.

It is felt that many different types of heating systems and controls may be simulated by this approach.

When the system status is "off" an optional frost protection thermostat may be selected to control the internal temperature.

Note that the calculated system load is that delivered to the building. As this sub-model is not a systems simulator it does not take into account the operating efficiency of the system. Since system demand and cycling may be part of the detailed data output however, efficiency corrections may possibly be made in a post-processor.

Associated routines are,
HTSYS , TSTAT , FROSTT , SYSDEL , STTDEL

3.1.8 Incidental Gains

The internal database allows incidental gains to be divided into four categories; lighting, occupants (physiological), small power, and others. Similar to the system heat sources the incidental loads may be specified as having both radiant and convective outputs and may be connected to various spaces and surfaces.

The 'others' category has no associated controlling structure built into HTB2 as it is intended for users extensions. Except for lighting the rest do not have models of behavior associated with them, they are designed to be varied by access via the scheduling diary. As mentioned, this diary control structure allows a very flexible control over the specification of incidental gains.

Lighting also operates in this manner, however a simple use model is also included. The lighting load may be described by "daytime" and "nighttime" loads, the choice between the two is made by reference to the external horizontal illumination (which in turn is determined from the horizontal irradiance). Although this is a very simplistic lighting model it serves as an illustration for future enhancement.

Associated routines are,
CASUAL , OCCUPD , LIGHTS , SPOWER , LGHTMD

3.1.9 Water Vapour Generation And Transport

Provision is made in the database to allow the water vapour content (in g/m3) of each space to be determined. The transportation of water vapour between spaces is treated in the same manner as convective heat transport, i.e. the net gain to a space is calculated. This is done within the ventilation routine. The external relative humidity is used as the main source of water vapour, but water vapour generation is allied to the processes of heating systems, and small power and occupants incidental gains. These are controlled through the diary schedule.

If water vapour calculations are selected then a watch is set on element surfaces to enable surface condensation risk to be calculated. Water vapour transmission through fabric components is not provided for, so that interstitial condensation is not considered, although preliminary work has shown that a transport process similar in structure to the finite difference fabric process can be developed.

Associated routines,
WATRSP , VENTL , OCCUPD , SPOWER

3.1.10 User Routine Link Points

Several linking points to dummy routines ('zippers') are provided throughout HTB2. Five are located at various points in the main HTB2 operating programme; these routines may be enabled at input or at run-time through the diary and would be called every timestep. Three further zippers are contained within the DIARY structure and may be called at any time, for a single time-step. A final zipper is located in the main ventilation routine, as it has been anticipated that this may be a major area of interest.

It is anticipated that these may be used for easy insertion of users routines, for example in controlling the 'others' incidental gains. It would be the responsibility of the users routine to access and alter the correct areas in the common database. It is hoped that these linking points may allow simple enhancement or inclusions to be made without major alteration of the existing model.

Associated routines are,
ZIPR1 , ZIPR2 , ZIPR3 , ZIPR4 , ZIPR5 , ZIPR6 , ZIPR7 , ZIPR8 , ZIPVNT

3.2 Input and Output

3.2.1 Input Data Organisation

The input data description of the building system has been sectioned to reflect the modules of the model calculations. Thus there are separate files specifying, for instance, the building layout, the materials used in construction, the heating system characteristics, etc. Each of these files are pointed to by an overall top-level input file, which also specifies the running options of the model. In this way standard or often used data sets may be built up and a simulation produced by mixing and matching.

To enhance the ease of use these data files are, as far as possible, in ordinary text (ascii) format and presented using english keywords rather than blocks of unreadable (by humans) numbers. Provision is made for extensive commenting within the data files so that, hopefully, easily understandable (and hence easily debugable), and traceable data files are produced.

Associated routines are,
INPUT , INIT

3.2.2 Diary Scheduling Input

The diary structure replaces the rigid hourly schedules of operation of aspects such as incidental gains and system control, usual to buldings models. It also, more fundamentally, can control the operation of the model itself by altering option flags or data variables during a calculation run. The diary operates by accessing the "common" data blocks of the process subroutines and altering variables via keywords.

The diary is organised into an 'index' and various 'pages'. Each page contains an ordered sequence of times (related to the time of day clock kept by the model) of events for a particular day, while the diary 'index' denotes which 'page' to use for which day of the simulation. A diary event is denoted as a command keyword , and parameters or values of the action to take. A typical diary entry may appear as;

```
       11:09:00   !SET HEATER TSTAT  £2 =  19.0
```

and this would be interpreted in the model as " alter the thermostat setting on heating system number 2 to 19.0 degree C at 11:09 precisely".

It is envisaged that this module will be continuously upgraded with use as the need for access to new variables is identified.

Associated routines are,

DIARY , DRYNXT

3.2.3 Output Procedures

Data output is available in two modes, the block output and the event log.

3.2.3.1 Block output reporting -

At intervals specified at start-up (hourly by default) a block report of average values describing the status of the simulation may be made. This data contains by default space air, radiant, and surface temperatures, humidities, heating loads, net incidental and solar gains etc. organised on a space by space basis. Further optional sections may be chosen which will record for instance meteorological, ventilation, or element surface data.

A futher, optional , block report file may be requested which will contain the temperature and heat flow profiles through selected elements at selected intervals (which need not be the same as the main report sequence).

Sufficient information is placed at the start of each output file so as to allow a trace on the input data to be made at future references to the output file.

Associated routines are;
REPORT , PRFOUT

3.2.3.2 Event logging -

Where short term detail is of interest , for example in the interactions of heating system controls and incidental or solar gains, simply decreasing the block reporting interval would produce a great amount of redundant information, little of which would be of interest. Here the logging of events, such as a temperature change greater than a specified amount, is of use. Data of interest, for example air temperatures, or system output, is identified and an event resolution, such as a change of greater than .5 degree or 100 watt, specified. The associated variables are watched at each time interval, and only recorded (as identification, time of occurrence, old value, and new value) when an event is identified. Thus data is available at the rate of change of the variable, and is sparse when data is changing slowly, i.e. overnight.

Associated routines are;
LOGGER

3.3 HTB2 Extra Features

Several novel features and capabilities are available which may be of use in research applications.

3.3.1 Virtual Spaces

Any element may be bounded on one side by what may be termed a virtual space, that is it does not exist as part of the simulation, but acts as an unlimited source or sink of heat. Basically it is a space that has a specified temperature that can exchange heat by fabric or ventilation transfer. As it is not modeled, however, its temperature will not change (unless, of course, it is changed through the diary). Such a structure will be useful in setting up constant boundary conditions or subjecting a modeled space to a known pattern of adjacent temperatures.

3.3.2 Virtual Partitions

Similarly a non-existent partition may be specified to zone a space. The heat transfer across this type of partition will depend soley on the air temperature differences across it. Given a reasonable model of convection a simple simulation of stratification may be possible. As the partition is non- existent it shows no impedance to radiative transfer.

3.3.3 Insolation Control

Each transparent element can have its associated transmission characteristic, and its shading templatet, altered at any time in the simulation, so as to simulate curtain or blind operation. The choice of the characteristics will be made via the diary.

3.3.4 Diary Control

The diary structure has further implications than simply allowing more flexible scheduling. It can for instance access the option and run variables of the model itself. While this might conceivably be disastrous if unwisely used, it may be useful in, for instance, turning on and off output at selected times.

The use of the diary to control the temperature of adjacent, 'unmodelled', spaces has already been mentioned, however a useful feature is the ability of the diary to inject a step change of heat content, or a pulse of heat, so as to study transient characteristics.

3.3.5 Geometric Considerations

HTB2 is not a spatial model in that it does not require three dimensional information about the building to be modelled. Thus there are no internal calculations of shape factors, shading, stratification layers, etc. This information may be supplied through external data files and so its derivation is independant of HTB2. This approach simplifies the internal calculations of HTB2 considerably and also allows investigation of physically unrealisable situations. The 'building' under simulation could thus be only one wall element subjected to imposed air temperatures.

3.3.6 User Link Points

At strategic areas in the calculations of HTB2, dummy 'zipper' routines are provided for possible linkage to users subroutines.

3.3.7 Routine Identification

Provision is made for each subroutine in HTB2 to write an identification string, including version information, to the default output stream on the first call of that subroutine. This is an option controlled by the main input file commands.

3.4 HTB2 Portability

HTB2 has been extensively rewritten to conform to ANSI standard FORTRAN-77 as far as possible. This should enable easier transportation between machines. Known exceptions to the standard are listed below. Also, where identifiable machine dependant processes (such as file manipulation) are isolated in separate subroutines.

HTB2 has been developed on VAX/VMS software for implementation on PRIME/PRIMOS, so that portability has been a particular requirement.

3.4.1 Use of INCLUDE statement

The nonstandard 'INCLUDE' statement is used to maintain the database commons in each subroutine.

3.4.2 Machine dependant routines

Operations which may require machine, compiler, or installation specific features are all located in separate subroutines within a machine-dependant library. At present the majority of these routines deal with the varying syntax requirments of the Fortran OPEN statement. Reading system time and date, and interpretation of Fortran error codes are also included here.

SECTION 2

HTB2 MODULE DESRIPTIONS AND DISCUSSION

A discussion of each routine used in the model HTB2 is given in this section.

HTB2 has been organised into distinct modules reflecting the conceptual breakdown of sub-systems in the building system. It is felt that this structure will allow both easier understanding of the operation, and easier maintenence or enhancement of the procedures, of the model.

The routines are grouped into chapters according to process group and follow, in the main, the modularisation as defined by the main operating programme HTBOS (presented in chapter 1). A diagrammatic relation of the routines is given in figure 1.

The documentation of each routine is comprised of a summary, a theoretical discussion of the algorithms or concepts used, source code, and a flowchart. Description and cross referencing of common or global variables used is contained in section 3 of this manual, while all local variables are explicitly declared at the start of each routine.

CHAPTER 1 MAIN PROGRAMME

1.1 HTBOS - MAIN PROGRAMME 1-1

CHAPTER 2 DATA INPUT

2.1 INPUT - READ TOP DATA FILE 2-2
2.2 DEFBLD - READ BUILDING DEFINITION FILE 2-5
2.3 RDCON - READ CONSTRUCTION DEFINITIONS 2-9
2.4 RDLAY - READ FABRIC LAYOUT DEFINITIONS 2-13
2.5 DEFSRV - READ SERVICES DEFINITION FILE 2-17
2.6 RDCOOL - READ COOLING SYSTEMS FILE 2-20
2.7 RDHTR - READ HEATING SYSTEMS FILE 2-23
2.8 RDLGT - READ LIGHTING SYSTEMS FILE 2-26
2.9 RDOCC - READ OCCUPANCY DEFINITION FILE 2-29
2.10 RDSPW - READ SMALL POWER SYSTEMS FILE 2-32
2.11 RDVENT - READ VENTILATION DEFINITION FILE . . . 2-35
2.12 RDVNT1 - READ VENT OPTION 1 DATA 2-38
2.13 RDVNT2 - READ VENT OPTION 2 DATA 2-41
2.14 RDVNT3 - READ VENT OPTION 3 DATA 2-44
2.15 RDVNT8 - READ VENT OPTION 8 DATA 2-47

CHAPTER 3 RUN INITIALISATION , RUN END

3.1 INIT - INITIALISE DATABASE 3-2
3.2 CHECKR - OUTPUT ALL INPUT DATA 3-5
3.3 INFO - STORE RUN INFORMATION TO FILE 3-8
3.4 TIDYUP - RUN END HOUSEKEEPING 3-11

CHAPTER 4 TIME KEEPING

4.1 CLOCK - MAINTAIN INTERNAL CLOCKS 4-2
4.2 NEWDAY - START OF NEW DAY 4-5

CHAPTER 5 DATABASE UPDATE

5.1 TEMPSP - CALCULATE NEW AIR TEMPERATURES 5-2
5.2 WATRSP - CALCULATE NEW AIR HUMIDITY 5-6
5.3 UPDATE - END OF TIMESTEP HOUSEKEEPING 5-9

CHAPTER 6 RADIANT TRANSFER

6.1 TEMPRD - STAR TEMPERATURE RADIANT TRANSFER . . . 6-2
6.2 MRTCLC - CALC MEAN RADIANT TEMPERATURE 6-7
6.3 EXCTRD - MATRIX EQUATION RADIANT EXCHANGE . . . 6-11

CHAPTER 7 METEOROLOGICAL DATABASE

7.1 METBAS - READ AND MAINTAIN MET DATABASE 7-2
7.2 EXTRHT - DETERMINE EXTERNAL CONVECTION COEFFS. . . 7-7
7.3 LNGWAV - DETERMINE EXTERNAL LONGWAVE LOSS . . . 7-11

CHAPTER 8 INSOLATION

8.1 SOLAR - DETERMINE SURFACE IRRADIANCE FROM SOLAR 8-2
8.2 SKYVW - DETERMINE ELEMENT OBSTRUCTION 8-6
8.3 TRFNCT - DETERMINE TRANSMISSION OF WINDOW . . . 8-11

CHAPTER 9 FABRIC HEAT TRANSFER

9.1 FABRIC - CALCULATED FABRIC TRANSFER 9-2
9.2 CAVTRN - CALCULATE CAVITY TRANSFER 9-9
9.3 HTCE - CHOSE EXTERNAL CONVECTIVE COEFFICIENT . 9-14
9.4 HTCF - CHOSE INTERNAL CONVECTIVE COEFFICIENT . 9-17
9.5 HTCX - CHOSE AIR-AIR CONVECTIVE COEFFICIENT . 9-20

CHAPTER 10 VENTILATION

10.1 VENTL - CALCULATE VENTILATION TRANSFER 10-2
10.2 VENT1 - CHOSE SPACE AIR CHANGE RATES 10-6
10.3 VENT2 - CHOSE SPACE TO SPACE FLOW PATTERN . . . 10-10
10.4 VENT3 - CALCULATE EMPIRICAL AIR CHANGE RATE . . 10-13
10.5 VENT8 - USE MULTICELL MATHEMATICAL MODEL . . . 10-17

CHAPTER 11 HEATING SYSTEM AND CONTROL RESPONSE

11.1 HTSYS - DETERMINE SYSTEM OUTPUT 11-2
11.2 TSTAT - DETERMINE THERMOSTAT OUTPUT 11-7
11.3 FROSTT - FROST PROTECTION T'STAT 11-13
11.4 SYSDEL - IMPOSE TIME DELAY ON SYSTEM OUTPUT . . 11-16
11.5 STTDEL - IMPOSE TIME DELAY ON T'STAT OUTPUT . . 11-19
11.6 CLSYS - CALCULATE COOLING LOAD 11-22

CHAPTER 12 INCIDENTAL GAINS

12.1 CASUAL - MANAGE INCIDENTAL GAIN SYSTEMS 12-2
12.2 OCCUPD - CALCULATE OCCUPANTS GAINS 12-5
12.3 SPOWER - CALCULATE GAINS FROM POWER SOURCES . . 12-8
12.4 LIGHTS - CALCULATE GAINS FORM LIGHT CIRCUITS . . 12-11
12.5 LGHTMD - LIGHTING LOAD MODEL 12-14

CHAPTER 13 DIARY SCHEDULE INPUT

13.1 DIARY - EXECUTE PENDING COMMAND 13-2
13.2 DRYNXT - QUEUE NEXT COMMAND 13-6
13.3 DRYIN - READ NEXT COMMAND FROM FILE 13-9

CHAPTER 14 OUTPUT PROCEDURES

14.1 REPORT - INTERVAL BLOCK OUTPUT 14-2
14.2 LOGGER - DATA EVENT LOGGER 14-5
14.3 PRFOUT - ELEMENT PROFILE OUTPUT 14-9

CHAPTER 15 USER LINK POINTS - ZIPPERS

15.1 ZIPR1 - LINK POINT 1 15-2
15.1 ZIPR2 - LINK POINT 2 15-2
15.1 ZIPR3 - LINK POINT 3 15-2
15.1 ZIPR4 - LINK POINT 4 15-2
15.1 ZIPR5 - LINK POINT 5 15-2
15.2 ZIPR6 - LINK POINT 6 15-9
15.2 ZIPR7 - LINK POINT 7 15-9
15.2 ZIPR8 - LINK POINT 8 15-9
15.3 ZPRVNT - VENTILATION LINK POINT 15-14
15.3 ZPRVNR - VENTILATION LINK POINT 15-14

CHAPTER 16 GENERAL UTILITY ROUTINES

16.1 CRASHR - ABORT RUN DUE TO ERRORS 16-2
16.2 RH2DP - CONVERT RH TO DEWPOINT 16-5
16.3 RH2VP - CONVERT RH TO VAPOUR PRESSURE 16-8
16.4 RH2WC - CONVERT RH TO WATER CONTENT 16-11
16.5 T2SVP - AIR TEMP TO SATURATED VAP. PRESSURE . . 16-14
16.6 VALDLN - TAKE DATA LINE FROM FILE 16-17
16.7 VP2WC - CONVERT VAP PRESSURE TO WATER CONTENT . 16-20
16.8 WC2RH - CONVERT WATER CONTENT TO RH 16-23
16.9 WHTSIZ - REPORT DATA TOTALS 16-26

CHAPTER 17 MACHINE/COMPILER SPECIFIC ROUTINES

17.1 DSTAMP - GET SYSTEM DATE AND TIME 17-2
17.2 FORERR - REPORT SYSTEM ERROR CODES 17-6
17.3 OPBLK1 - OPEN REPORT FILE 17-10
17.4 OPBLK2 - OPEN PROFILE FILE 17-14
17.5 OPLGR1 - OPEN EVENT LOGGER FILE 17-18
17.6 OPNBLD - OPEN BUILDINGS FILE 17-22
17.7 OPNCON - OPEN CONSTRUCTION FILE 17-26
17.8 OPNDRY - OPEN DIARY PAGE FILE 17-30
17.9 OPNDYL - OPEN DIARY LIST FILE 17-34
17.10 OPNHTR - OPEN HEATING SYSTEM FILE 17-38

17.11 OPNINF - OPEN INFORMATION FILE 17-42
17.12 OPNLAY - OPEN FABRIC LAYOUT FILE 17-46
17.13 OPNLGT - OPEN LIGHTING SYSTEM FILE 17-50
17.14 OPNMAT - OPEN MATERIALS LIBRARY FILE 17-54
17.15 OPNMET - OPEN METEOROLOGICAL FILE 17-58
17.16 OPNOCC - OPEN OCCUPANCY FILE 17-62
17.17 OPNSPW - OPEN SMALL POWER FILE 17-66
17.18 OPNSRV - OPEN SERVICES FILE 17-70
17.19 OPNTOP - OPEN TOP-LEVEL FILE 17-74
17.20 OPNVNT - OPEN VENTILATION FILE 17-78
17.14 OPUMAT - OPEN USERS MATERIALS FILE 17-82

SECTION 2 INDEX BY ROUTINE NAME

CASUAL - MANAGE INCIDENTAL GAIN SYSTEMS 12-2
CAVTRN - CALCULATE CAVITY TRANSFER 9-9
CHECKR - OUTPUT ALL INPUT DATA 3-5
CLOCK - MAINTAIN INTERNAL CLOCKS 4-2
CLSYS - CALCULATE COOLING LOAD 11-22
CRASHR - ABORT RUN DUE TO ERRORS 16-2

DIARY - EXECUTE PENDING COMMAND 13-2
DEFBLD - READ BUILDING DEFINITION FILE 2-5
DEFSRV - READ SERVICES DEFINITION FILE 2-17
DRYIN - READ NEXT COMMAND FROM FILE 13-9
DRYNXT - QUEUE NEXT COMMAND 13-6
DSTAMP - GET SYSTEM DATE AND TIME 17-2

EXCTRD - MATRIX EQUATION RADIANT EXCHANGE . . . 6-11
EXTRHT - DETERMINE EXTERNAL CONVECTION COEFFS. . . 7-7

FABRIC - CALCULATED FABRIC TRANSFER 9-2
FORERR - REPORT SYSTEM ERROR CODES 17-6
FROSTT - FROST PROTECTION T'STAT 11-13

HTBOS - MAIN PROGRAMME 1-1
HTCE - CHOSE EXTERNAL CONVECTIVE COEFFICIENT . 9-14
HTCF - CHOSE INTERNAL CONVECTIVE COEFFICIENT . 9-17
HTCX - CHOSE AIR-AIR CONVECTIVE COEFFICIENT . 9-20
HTSYS - DETERMINE SYSTEM OUTPUT 11-2

INIT - INITIALISE DATABASE 3-2
INFO - STORE RUN INFORMATION TO FILE 3-8
INPUT - READ TOP DATA FILE 2-2

LIGHTS - CALCULATE GAINS FORM LIGHT CIRCUITS . . 12-11
LGHTMD - LIGHTING LOAD MODEL 12-14
LOGGER - DATA EVENT LOGGER 14-5
LNGWAV - DETERMINE EXTERNAL LONGWAVE LOSS . . . 7-11

METBAS - READ AND MAINTAIN MET DATABASE 7-2
MRTCLC - CALC MEAN RADIANT TEMPERATURE 6-7

NEWDAY - START OF NEW DAY 4-5

OCCUPD - CALCULATE OCCUPANTS GAINS 12-5
OPBLK1 - OPEN REPORT FILE 17-10
OPBLK2 - OPEN PROFILE FILE 17-14
OPLGR1 - OPEN EVENT LOGGER FILE 17-18
OPNBLD - OPEN BUILDINGS FILE 17-22
OPNCON - OPEN CONSTRUCTION FILE 17-26
OPNDRY - OPEN DIARY PAGE FILE 17-30
OPNDYL - OPEN DIARY LIST FILE 17-34
OPNHTR - OPEN HEATING SYSTEM FILE 17-38
OPNINF - OPEN INFORMATION FILE 17-42

OPNLAY - OPEN FABRIC LAYOUT FILE 17-46
OPNLGT - OPEN LIGHTING SYSTEM FILE 17-50
OPNMAT - OPEN MATERIALS LIBRARY FILE 17-54
OPNMET - OPEN METEOROLOGICAL FILE 17-58
OPNOCC - OPEN OCCUPANCY FILE 17-62
OPNSPW - OPEN SMALL POWER FILE 17-66
OPNSRV - OPEN SERVICES FILE 17-70
OPNTOP - OPEN TOP-LEVEL FILE 17-74
OPNVNT - OPEN VENTILATION FILE 17-78
OPUMAT - OPEN USERS MATERIALS FILE 17-82

PRFOUT - ELEMENT PROFILE OUTPUT 14-9

RDCON - READ CONSTRUCTION DEFINITIONS 2-9
RDLAY - READ FABRIC LAYOUT DEFINITIONS 2-13
RDCOOL - READ COOLING SYSTEMS FILE 2-20
RDHTR - READ HEATING SYSTEMS FILE 2-23
RDLGT - READ LIGHTING SYSTEMS FILE 2-26
RDOCC - READ OCCUPANCY DEFINITION FILE 2-29
RDSPW - READ SMALL POWER SYSTEMS FILE 2-32
RDVENT - READ VENTILATION DEFINITION FILE . . . 2-35
RDVNT1 - READ VENT OPTION 1 DATA 2-38
RDVNT2 - READ VENT OPTION 2 DATA 2-41
RDVNT3 - READ VENT OPTION 3 DATA 2-44
RDVNT8 - READ VENT OPTION 8 DATA 2-47
REPORT - INTERVAL BLOCK OUTPUT 14-2
RH2DP - CONVERT RH TO DEWPOINT 16-5
RH2VP - CONVERT RH TO VAPOUR PRESSURE 16-8
RH2WC - CONVERT RH TO WATER CONTENT 16-11

SKYVW - DETERMINE ELEMENT OBSTRUCTION 8-6
SOLAR - DETERMINE SURFACE IRRADIANCE FROM SOLAR 8-2
SPOWER - CALCULATE GAINS FROM POWER SOURCES . . 12-8
STTDEL - IMPOSE TIME DELAY ON T'STAT OUTPUT . . 11-19
SYSDEL - IMPOSE TIME DELAY ON SYSTEM OUTPUT . . 11-16

TEMPRD - STAR TEMPERATURE RADIANT TRANSFER . . . 6-2
TEMPSP - CALCULATE NEW AIR TEMPERATURES 5-2
TIDYUP - RUN END HOUSEKEEPING 3-11
TRFNCT - DETERMINE TRANSMISSION OF WINDOW . . . 8-11
TSTAT - DETERMINE THERMOSTAT OUTPUT 11-7
T2SVP - AIR TEMP TO SATURATED VAP. PRESSURE . . 16-14

UPDATE - END OF TIMESTEP HOUSEKEEPING 5-9

VALDLN - TAKE DATA LINE FROM FILE 16-17
VENTL - CALCULATE VENTILATION TRANSFER 10-2
VENT1 - CHOSE SPACE AIR CHANGE RATES 10-6
VENT2 - CHOSE SPACE TO SPACE FLOW PATTERN . . . 10-10
VENT3 - CALCULATE EMPIRICAL AIR CHANGE RATE . . 10-13
VENT8 - USE MULTICELL MATHEMATICAL MODEL . . . 10-17
VP2WC - CONVERT VAP PRESSURE TO WATER CONTENT . 16-20

WATRSP - CALCULATE NEW AIR HUMIDITY 5-6
WC2RH - CONVERT WATER CONTENT TO RH 16-23
WHTSIZ - REPORT DATA TOTALS 16-26

ZIPR1 - LINK POINT 1 15-2
ZIPR2 - LINK POINT 2 15-2
ZIPR3 - LINK POINT 3 15-2
ZIPR4 - LINK POINT 4 15-2
ZIPR5 - LINK POINT 5 15-2
ZIPR6 - LINK POINT 6 15-9
ZIPR7 - LINK POINT 7 15-9
ZIPR8 - LINK POINT 8 15-9
ZPRVNT - VENTILATION LINK POINT 15-14
ZPRVNR - VENTILATION LINK POINT 15-14

CHAPTER 1

MAIN PROGRAMME

1.1 HTBOS

GENERAL DESCRIPTION :

HTBOS is the overall operating programme of the model HTB2. It manages the operation of the data input, the simulation, and the output modules. The operation of a particular module is governed by a selection flag, ie. a particular module may be passed over in the simulation according to the state of option variables, or by its location within a timeframe, so that different processes may be calculated at different time intervals.

THEORETICAL BASIS :

The overall approach of HTB2 is best summarised by the partitioning of time and of process as discussed in section 1 of this manual.

To summarise time is partitioned into discrete intervals over which all boundary conditions are held to remain constant. As the same boundary conditions are seen by each distinct process of the modelled system, each process may be held to act independantly over a time interval, thus partitioning processes. Interactions between processes occur over multiple time intervals as the accumulated effects of each process is used to determine the new boundary conditions for the following time interval.

Within a particular time interval, the order of the process sub-models is not significant, as within the partitioning framework they are determined circularily (fig 1.1). They are placed in HTBOS in what is felt to be a reasonably intuitive order.

Most modules are optional in the calculation chain so that particular aspects for investigation may be isolated. The most common usage of HTB2 will of course require all modules in operation. A particular module is chosen for inclusion in the simulation by option flags set in the data input sequence.

CHAPTER 2

DATA INPUT

These routines deal with the reading and interpretation of the input data files.

INPUT - read top-level data file 2-2
  DEFBLD - read building definition file . . . 2-5
    RDCON - read construction definitions . . 2-9
    RDLAY - read fabric layout definitions . . 2-13
  DEFSRV - read services definition file . . . 2-17
    RDCOOL - read cooling systems file 2-20
    RDHTR - read heating systems file 2-23
    RDLGT - read lighting systems file 2-26
    RDOCC - read occupancy definition file . . 2-29
    RDSPW - read small power systems file . . 2-32
    RDVENT - read ventilation definition file . 2-35
      RDVNT1 - read vent option 1 data 2-38
      RDVNT2 - read vent option 2 data 2-41
      RDVNT3 - read vent option 3 data 2-44
      RDVNT8 - read vent option 8 data 2-47

2.1 INPUT - read top-level data file - level 1

CALLED FROM : HTBOS

CALL FORMAT : CALL INPUT

CALLS TO : DEFBLD , DEFSRV , FORERR , OPBLK1 , OPBLK2 , OPLGR1 , OPNDRY , OPNDYL , OPNMET , OPNTOP , VALDLN , WHTSIZ
CALL FREQUENCY : once

GENERAL DESCRIPTION:

INPUT is the general data input module. It set most variables to initial or default values and processes data files to load the users building and services descriptions.

INPUT interprets command lines from the top level input file. This file contains commands to set model parameters, such as the run length, and run options, such as the choice of output procedures. It also contains pointers to further files to be read for the process data, such as the building fabric description, or the heating systems specification. INPUT manages the routines which interpret these individual process data files.

The general format of commands to INPUT is;
Command Action { = Parameter} ,
where command is the data type, i.e. !SET , !ENABLE , !OUTPUT,
action is the object of the command, i.e. RUNLENGTH , FABRIC ,
parameter is an numeric or string arguement for the object.
A complete list of input'commands and their syntax is given in the user manual.

I/O:

Identification to default device (unit '*') if requested.
Error and information messages to default device.
Formatted, sequential character input from top-level file, on unit TOPUNT.
Formatted data input from 'internal files'.

ERROR TRAPPING:

Reports error or warning messages on unknown or mistyped commands.
Reports errors on handling files.
Reports inconsistencies and significant omissions in option choices.
Detection of errors halts program at end of routine.

2.2 DEFBLD - read building definition file - level 2

CALLED FROM : INPUT

CALL FORMAT : CALL DEFBLD(FILE,IERR)
: FILE - char - filename to read
: IERR - intg - returned error code;
: 0 - success
: -1 - errors detected

CALLS TO : FORERR , OPNBLD , OPNMAT , OPUMAT , RDCON , RDLAY , VALDLN
CALL FREQUENCY : once

GENERAL DESCRIPTION:

DEFBLD interprets commands taken from the file nominated in the top-level data file. These commands define the building site location, internal volumes and spaces, and pointers to further data files containing details on fabric construction and partitioning.

DEFBLD manages the routines which interpret these files, precalculates certain physical constants of the building, and cross-references concepts such as spaces, radiant zones, and surfaces. DEFBLD also checks for numerical stability in the finite difference process, and determines the optimum division of materials into equal distance finite difference slices.

THEORETICAL BASIS:

The numerical stability of the finite difference process is determined from the well known relation for internal explicit finite difference transport,

$$0.5 \geq \frac{dT * k}{D * C * dX^2}$$

where dT is the timestep in seconds,
dX is the material thickness in meters,
k is the thermal conductivity W/m/oK
D is the material density kg/m3
and C is the material specific heat J/kg/oK.

Analysis of the finite difference process used in the routine FABRIC has indicated that this, rather than surface transport, is the critical consideration in stability in this implementation.

DEFBLD uses this relation, with the timestep requested in the top-level file, to determine the maximum (within a limited range) number of equal distance slices required for each construction type used. If stabilty cannot be achieved for the requested timestep an error message is produced and the programme stopped at the end of the input stage. Alternately, the number of slices may be explicitly stated in the input file in which case stability checking is not done.

I/O:

Identification to default device if required.
Error and information messages to default device.
Formatted, sequential character input from data files on unit INPUNT.
Formatted data input from 'internal files'.
Formatted direct access input from materials library on unit INPUNT.

ERROR TRAPPING:

Reports error or warning messages on unknown or mistyped commands.
Reports errors in handling files.
Reports inability to maintain stable timestep.
Detection of errors halts program at end of input stage.

2.3 RDCON - read construction definitions - level 3

CALLED FROM : DEFBLD

CALL FORMAT : CALL RDCON(FILE,IERR)
: FILE - char - filename to read
: IERR - intg - returned error code;
: 0 - success
: -1 - errors detected

CALLS TO : FORERR , OPNCON , VALDLN

CALL FREQUENCY : once

GENERAL DESCRIPTION:

RDCON interprets commands from the construction file nominated in the building definition file. These commands define the properties of construction types, and the transmission properties of window types, which may be used in the fabric partitioning of the building.

A construction type is defined as a sequence of materials and thicknesses which make up a distinct partitioning element. The order of the layers specified is assumed to proceed from the 'left' surface to the 'right' surface (refer to the user manual for a discussion of 'left' and 'right' terminology). The materials used in the layers are identified by code numbers which refer to thermodynamic properties held in a library file (which has been nominated in the building definition file, or in this file). An alternate materials file is also available by prefixing the code number with the character '@'. Construction types are assigned sequential code numnbers as they are defined, these codes are used in the definition of the fabric layout.

The details specified for each layer of a construction are the material code, the layer thickness, and the number of slices desired for the finite-difference calculations. This latter may be specified as 0, which will require DEFBLD to make the best choice. For transparent constructions only, a fourth parameter is needed which defines the absorption of this layer, as a proportion of the total absorption of the construction.

A window type is defined as a specification of the direct transmissions and absorptances for incidence angles in 10 degree steps, and for net diffuse transmission and absorptance. Note that the absorptance in particular is specified as the total for the window structure, the partitioning of this absorption to the distinct layers of a construction is specified in the construction definition. A window type type is defined and referred to by a character code name.

I/O:

Identification to default device if requested.
Error and information messages to default device.
Formatted sequential character input from nominated file, on unit INPUNT.

ERROR TRAPPING:

Reports errors on unknown, mistyped, or badly ordered commands.
Detection of erroes halts program after input stage.

2.4 RDLAY - read building layout definition - level 3

CALLED FROM : DEFBLD

CALL FORMAT : CALL RDLAY(FILE,IERR,ISPMAX,IZNMAX)
: FILE - char - filename to read
: IERR - intg - returned error code
0 - success
-1 - errors detected
: ISPMAX - intg - maximum spaces used in layout
: IZNMAX - intg - maximum zones used in layout

CALLS TO : FORERR , OPNLAY , VALDLN

CALL FREQUENCY : once

GENERAL DESCRIPTION:

RDLAY interprets commands in the building layout definition file which describe the partitioning of the building spaces by defining fabric elements. The parameters defined include construction type code used in a partitioning element, the surface space and radiant zone connections, areas, orientations, surface characteristics, etc.

RDLAY also defines shading masks for external elements.

An element is defined as being a connection between spaces (the exterior is a special space), and may be physical, as in a wall, or virtual, as in an imaginary subdivision of a large volume. An element is made of only one construction, and each surface may connect to only one space. An elements surface is considered to be of uniform temperature and heat flux.

Details required for each element are;

- construction code (as defined in RDCON)
- element area
- orientation in degrees clockwise from south
- tilt in degrees clockwise from horizontal to normal of 'left' surface
- connecting spaces and radiant zones
- surface emissivities and absorptions
- shade mask, and if transparent, window type and sunpatching data
- adjacent ground reflectance

Many of these can default to preset values where appropriate. See the user manual for further details on data requirements.

A shading mask is a template of the sky vault, in 10 degree sectors, as seen from a surface. Each sector has a transmission factor associated with it, thus site obstruction shading, or overall veiling may be specified. These masks are defined in the layout file, and are also connected to particular elements within this file. Shading masks are used to determine the solar irradiance falling on any external surface, and thus to determine the shaded solar transmission of transparent elements. Masks are of importance only to external connecting elements. There is one predefined mask, named 'NONE', which represents unobstructed views. This is the default mask chosen if none are specified.

I/O:

Identification to default device if requested.
Error and information messages to default device.
Formatted sequential character input from nominated file, on unit INPUNT.

ERROR TRAPPING:

Reports errors on unknown, mistyped, or badly ordered commands.
Reports abscence of necessary specifications.
Detection of errors halts program after input stage.

Virtual Spaces ?

2.5 DEFSRV - read services definition file - level 2

```
CALLED FROM       : INPUT

CALL FORMAT       : CALL DEFSRV(FILE,IERR)
                    : FILE - char - filename to read
                    : IERR - intg - returned error code;
                                  :  0 - success
                                  : -1 - errors detected

CALLS TO          : OPNSRV , RDCOOL , RDHTR , RDLGT , RDOCC , RDSPW ,
                    RDVENT , VALDLN
CALL FREQUENCY    : once
```

GENERAL DESCRIPTION:

DEFSRV interprets commands in the services definition file selected in the top-level file. This file contains commands which point to further definition files for each service process; cooling systems, heating systems, lighting systems, occupancy parameters, small power systems, and ventilation. Data exists for any of these systems only if they have been nominated in the top-level file. Note that the corresponding processes must also be enabled in the top-level file before any calculations concerning them will be undertaken.

DEFSRV manages the routines which read the services data files.

I/O:

Indentification to default device if requested.
Error and information messages to default device.
Formatted, sequential character input from definition file on unit INPUNT.

ERROR TRAPPING:

Reports error or warning messages on unknown or mistyped commands.
Detection of errors halts program at end of input stage.

2.6 RDCOOL - read cooling systems file - level 3

CALLED FROM : DEFSRV

```
CALL FORMAT       : CALL RDCOOL(FILE,IERR)
                        :  FILE - char - filename to read
                        :  IERR - intg - returned error code
                                          0 - success
                                         -1 - errors detected
```

CALLS TO : CRASHR

CALL FREQUENCY : once

GENERAL DESCRIPTION:

At present RDCOOL is a dummy subroutine which will abort the run if called. A cooling load submodel is to be included in HTB2 at a later date, for which this will be the input routine.

I/O:

Identification to default device if requested.

ERROR TRAPPING:

Programme halted if routine called.

2.7 RDHTR - read heating systems file -level 3

```
CALLED FROM      : DEFSRV

CALL FORMAT      : CALL RDHTR(FILE,IERR)
                      : FILE - char - filename to read
                      : IERR - intg - returned error code
                                    0 - success
                                   -1 - errors detected

CALLS TO         : OPNHTR

CALL FREQUENCY   : once
```

GENERAL DESCRIPTION:

RDHTR is an interim input routine for the heating system submodel. It accepts data in a fixed sequential numeric format to describe the characteristics of the simulated heating systems. There are position flags inserted in the expected data stream to allow some detection of missing data records. There is no provision for default values, all data must be explicitly declared. For further details of the heating system sub-model, and the data required, see the module HTSYS.

I/O:

Identification to default device if requested.
Error and information messages to default device.
Formatted sequential numeric input from nominated file, on unit INPUNT.

ERROR TRAPPING:

Reports unreadable data records, data records out of position.
Detection of errors halts programme at end of input phase.

2.8 RDLGT - read lighting systems file -level 3

CALLED FROM : DEFSRV

```
CALL FORMAT       : CALL RDLGT(FILE,IERR)
                        :  FILE - char - filename to read
                        :  IERR - intg - returned error code
                                            0 - success
                                           -1 - errors detected
```

CALLS TO : OPNLGT

CALL FREQUENCY : once

GENERAL DESCRIPTION:

RDLGT is an interim input routine for the lighting system submodel. It accepts data in a fixed sequential numeric format to describe the characteristics of the simulated lighting circuits. There are position flags inserted in the expected data stream to allow some detection of missing data records. There is no provision for default values, all data must be explicitly declared. Further details on the lighting sub-model, and the required data, is found in the module CASUAL.

I/O:

Identification to default device if requested.
Error and information messages to default device.
Formatted sequential numeric input from nominated file, on unit INPUNT.

ERROR TRAPPING:

Reports unreadable data records, data records out of position.
Detection of errors halts programme at end of input phase.

2.9 RDOCC - read occupancy definition file -level 3

```
CALLED FROM      : DEFSRV

CALL FORMAT      : CALL RDOCC(FILE,IERR)
                     :  FILE - char - filename to read
                     :  IERR - intg - returned error code
                                       0 - success
                                      -1 - errors detected

CALLS TO         : OPNOCC

CALL FREQUENCY   : once
```

GENERAL DESCRIPTION:

RDOCC is an interim input routine for the occupancy gains submodel. It accepts data in a fixed sequential numeric format to describe the metabolic levels and occupancy rates for specific spaces. There are position flags inserted in the expected data stream to allow some detection of missing data records. There is no provision for default values, all data must be explicitly declared.

I/O:

Identification to default device if requested.
Error and information messages to default device.
Formatted sequential numeric input from nominated file, on unit INPUNT.

ERROR TRAPPING:

Reports unreadable data records, data records out of position.
Detection of errors halts programme at end of input phase.

2.10 RDSPW - read small power systems file -level 3

CALLED FROM : DEFSRV

```
CALL FORMAT        : CALL RDSPW(FILE,IERR)
                         : FILE - char - filename to read
                         : IERR - intg - returned error code
                                          0 - success
                                         -1 - errors detected
```

CALLS TO : OPNSPW

CALL FREQUENCY : once

GENERAL DESCRIPTION:

RDSPW is an interim input routine for the small power system submodel. It accepts data in a fixed sequential numeric format to describe the characteristics of the simulated heating systems. There are position flags inserted in the expected data stream to allow some detection of missing data records. There is no provision for default values, all data must be explicitly declared.

I/O:

Identification to default device if requested.
Error and information messages to default device.
Formatted sequential numeric input from nominated file, on unit INPUNT.

ERROR TRAPPING:

Reports unreadable data records, data records out of position.
Detection of errors halts programme at end of input phase.

2.11 RDVENT - read ventilation definition file -level 3

CALLED FROM : DEFSRV

CALL FORMAT : CALL RDVENT(FILE,IERR)
: FILE - char - filename to read
: IERR - intg - returned error code
0 - success
-1 - errors detected

CALLS TO : FORERR , OPNVNT , RDVNT1 , RDVNT2 , RDVNT3 , RDVNT8 , ZPRVNR

CALL FREQUENCY : once

GENERAL DESCRIPTION:

RDVENT is an interim input routine for the ventilation submodel. It accepts data in a fixed sequential numeric format to select the ventiliation calculation option required, and manages the input routines to read the specific data for the option chosen.

I/O:

Identification to default device if requested.
Error and information messages to default device.
Formatted sequential numeric input from nominated file, on unit INPUNT.

ERROR TRAPPING:

Reports unreadable data records, data records out of position.
Detection of errors halts programme at end of input phase.

2.12 RDVNT1 - read vent option 1 data -level 4

```
CALLED FROM      : RDVENT

CALL FORMAT      : CALL RDVNT1(IERR)
                     :  IERR - intg - returned error code
                                   0 - success
                                  -1 - errors detected

CALLS TO         : FORERR

CALL FREQUENCY   : once
```

GENERAL DESCRIPTION:

RDVNT1 is an interim input routine for the ventilation option 1 submodel. It accepts data in a fixed sequential numeric format to describe the air change rates of selected spaces. There are position flags inserted in the expected data stream to allow some detection of missing data records. Any space not defined defaults to zero ventilation.

I/O:

Identification to default device if requested.
Error and information messages to default device.
Formatted sequential numeric input from nominated file, on unit INPUNT.

ERROR TRAPPING:

Reports unreadable data records, data records out of position.
Detection of errors halts programme at end of input phase.

2.13 RDVNT2 - read vent option 2 data -level 4

```
CALLED FROM       : RDVENT

CALL FORMAT       : CALL RDVNT2(IERR)
                        :  IERR - intg - returned error code
                                         0 - success
                                        -1 - errors detected

CALLS TO          : FORERR

CALL FREQUENCY    : once
```

GENERAL DESCRIPTION:

RDVNT2 is an interim input routine for the vent option 2 submodel. It accepts data in a fixed sequential numeric format to describe the space-space air flow characteristics of the simulated building. There are position flags inserted in the expected data stream to allow some detection of missing data records. Any undefined connection defaults to zero airflow.

I/O:

Identification to default device if requested.
Error and information messages to default device.
Formatted sequential numeric input from nominated file, on unit INPUNT.

ERROR TRAPPING:

Reports unreadable data records, data records out of position.
Detection of errors halts programme at end of input phase.

2.14 RDVNT3 - read vent option 3 data - level 4

CALLED FROM : RDVENT

CALL FORMAT : CALL RDVNT3(IERR)
: IERR - intg - returned error code
0 - success
-1 - errors detected

CALLS TO : FORERR

CALL FREQUENCY : once

GENERAL DESCRIPTION:

RDVNT3 is an interim input routine for the ventilation option 3 submodel. It accepts data in a fixed sequential numeric format to describe the parameters of a emperical equation set. There are position flags inserted in the expected data stream to allow some detection of missing data records. There is no provision for default values, all data must be explicitly declared.

I/O:

Identification to default device if requested.
Error and information messages to default device.
Formatted sequential numeric input from nominated file, on unit INPUNT.

ERROR TRAPPING:

Reports unreadable data records, data records out of position.
Detection of errors halts programme at end of input phase.

2.15 RDVNT8 - read vent option 8 data - level 4

CALLED FROM : RDVENT

CALL FORMAT : CALL RDVNT8(IERR)
: IERR - intg - returned error code
0 - success
-1 - errors detected

CALLS TO : CRASHR

CALL FREQUENCY : once

GENERAL DESCRIPTION:

RDVNT8 is a dummy routine for later inclusion of a dynamic ventilation submodel. Calling this routine will halt the programme run.

I/O:

Identification to default device if requested.
Error and information messages to default device.

ERROR TRAPPING:

Calling routine halts programme immediately.

CHAPTER 3

RUN INITIALISATION, RUN END

This routine deals with the initialisation of any database variables not set up in the input stage, and with housekeeping at the end of a run.

INIT - initialise database 3-2
  CHECKR - output all input data 3-5
  INFO - store run information to file . . 3-8
TIDYUP - run end housekeeping 3-11

3.1 INIT - initialise database - level 1

CALLED FROM : HTBOS

CALL FORMAT : CALL INIT

CALLS TO : CHECKR , DSTAMP , INFO

CALL FREQUENCY : once

GENERAL DESCRIPTION

INIT initialises parts of the database which where not involved in the input stage, in particular accumulators and temperatures are set to predetermined initial values. At the start of a simulation all fabric and space temperatures are set to a default value, TREF.

The date and time of the simulation run is determined from the computing system. Various parameters, such as run options and input file names, are output to an information file for later access by post-processing programmes, or for tracing run histories.

As an option, all input data may be output in a formatted file for checking validity (n.b. this routine is still in preparation).

I/O:

Identification to default device if required.

3.2 CHECKR - output all input data - level 2

CALLED FROM : INIT

CALL FORMAT : CALL CHECKR

CALLS TO : -

CALL FREQUENCY : once

GENERAL DESCRIPTION:

CHECKR is at present a dummy routine. It is intended to output, in formatted, easily readable style, all data read in the input stage so that validity checking may be carried out.

I/O:

Identification to default device if requested.
Error messages to default device.

ERROR TRAPPING:

The dummy routine will halt execution of programme if called.

3.3 INFO - store run information to filr -level 2

CALLED FROM : INIT

CALL FORMAT : CALL INFO

CALLS TO : OPNINF

CALL FREQUENCY : once

GENERAL DESCRIPTION:

INFO records filenames , run options and data cross-references which may be of use to post-processing programmes.

I/O:

Identification to default device if requested.
Formatted, sequential output to file on unit INFUNT.

ERROR TRAPPING:

Run aborted if information file cannot be opened.

3.4 TIDYUP - run end housekeeping - level 1

CALLED FROM : HTBOS

CALL FORMAT : CALL TIDYUP

CALLS TO : -

CALL FREQUENCY : once

GENERAL DESCRIPTION:

TIDYUP closes down a run in an orderly manner. At present this just entails correctly closing all data files which may be open.

I/O:

Identification to default device if required.

CHAPTER 4

TIME KEEPING

These routines deal with maintaining the time-of-day clocks in HTB2, and with managing any strictly time dependant routines.

CLOCK - maintain internal clocks 4-2
NEWDAY - start of new day 4-5

4.1 CLOCK - maintain internal clocks - level 1

CALLED FROM : HTBOS

CALL FORMAT : CALL CLOCK

CALLS TO : -

CALL FREQUENCY : every time-step

GENERAL DESCRIPTION:

CLOCK maintains;

1. the time of day clocks (seconds past midnight and hh:mm:ss.s),
2. the current date as counted from the initial date,
3. the day of week as counted from the initial day,
4. the day count in the year,
5. the pointers to the current location in storage rings,
6. the flags indicating midnight and the end of the run.

CLOCK is called every timestep, and increments the time counters by the time-step length.

Leap years are considered for years divisible by 4.

Alternate time systems (i.e. BST) are entered by specifiying the start and stop dates (as day count in year). One hour is added to the time at one o'clock on the start day and subtracted on the stop day.

Hours run are totalised and when the run length is exceeded the end of run flag is set.

I/O:

Identification to default device if required.

4.2 NEWDAY - start of new day - level 1

CALLED FROM : HTBOS

CALL FORMAT : CALL NEWDAY

CALLS TO : CRASHR , DRYNXT , FORERR , OPNDRY , VALDLN

CALL FREQUENCY : once a day, after midnght passed

GENERAL DESCRIPTION:

NEWDAY manages any processes necessary at the start of each day.

This includes altering the next action times for data input and output. These times are accumulations of the action interval in seconds, however the model time (in seconds) is reset to zero at midnight. This routine therefore normalises the action times to 0 -86400 seconds (0 -1 day).

This routine also manages the selection of the diary page files. A new file is opened at the tart of each day, for use by the diary routine.

I/O:

Identification to default device on request.
Error, warning, and information messages to default device.
Formatted sequential input on diary list unit DYLUNT.

ERROR TRAPPING:

Errors on handling diary files will abort run.

CHAPTER 5

DATABASE UPDATE

These routines determine the new conditions resulting from the accumulated effects of the previous timestep calculations. General housekeeping is also included here.

TEMPSP - clculate new air temperature . . . 5-2
WATRSP - calculate new air humidity 5-6
UPDATE - end of timestep housekeeping . . . 5-9

5.1 TEMPSP - calculate new air temperature - level 1

CALLED FROM : HTBOS

CALL FORMAT : CALL TEMPSP

CALLS TO : -

CALL FREQUENCY : every time-step

GENERAL DESCRIPTION:

TEMPSP applies the convective gains calculated from each transfer process (ie. fabric, ventilation, heating plant) to the total heat content of the space air and calculates a new air temperature from that new heat content.

THEORETICAL BASIS:

Each modelled space has a heat content related to the air temperature, defined as

$$H = (Ta - Tr) * V$$

,or

$$Ta = \frac{H}{V} + Tr ,$$

where H is the heat content of space air (j/m3)
Ta is the space air temperature (oC)
Tr is the reference temperature (model parameter - oC)
and V is the air heat capacity of the space (j/m3/oC), (precalculated on input from volume of space).

This air heat content is relative to a reference temperature TREF (Tr above) which should approximate the average temperature expected, normally 10oC, so as to decrease internal roundoff errors.

Each transfer process, ie. fabric, ventilation, or heating plant, will have produced as a reaction to the condition of the last timestep, a convective gain (in watts) for each space. These gains are at this stage totalised and applied to the space, multiplied by the process time step to determine heat change, to calculate the air temperatures for the current timestep. Once this has been applied to the space the process gains are reset to zero for the next round of calculations.

Unmodelled spaces (virtual spaces) do not take part in these calculations, so that their temperatures are unaffected by the process results, and remain constant unless altered through the diary structure. In this way virtual spaces can act as ideal sources or sinks of heat.

If there are no modelled spaces in the simulation, then this routine simply returns.

I/O:

Identification to default device if requested.

5.2 WATRSP - calculate new air humidity - level 1

CALLED FROM : HTBOS

CALL FORMAT : CALL WATRSP

CALLS TO : RH2DP , WC2RH

CALL FREQUENCY : every timestep, if water calculation enabled

GENERAL DESCRIPTION:

WATRSP determines the new air water content, humidity and dewpoint of a modelled space from the net effects of the previous timestep transfer processes.

THEORETICAL BASIS:

Each space has an associated water content in g/m3 of air. This water is available for transfer by ventilation, and may be added to, or removed by, other processes such as heating systems or small power sources.

As a result of the previous timestep, each process will have produced a net water gain , in g/s, for each modelled space. This routine collates these gains and applies them, in a similar manner as heat is handled in TEMPSP, to produce a new water content of the space. From this value, humidity and dewpoint for the space air are determined.

Virtual, unmodelled spaces are not included in these calculation so their water content remains constant unless altered thought the diary.

I/O:

Identification to default device if requested.

5.3 UPDATE - end of timestep houskeeping - level 1

CALLED FROM : HTBOS

CALL FORMAT : CALL UPDATE

CALLS TO : CRASHR

CALL FREQUENCY : every timestep

GENERAL DESCRIPTION:

UPDATE manages any end-of-timestep housekeeping.

At present this mainly concerns resetting single shot logical flags, such as those used to force particular calculations during the last timestep. This routine is also used to provide 4th power surface temperatures for possible use by radiant transfer routines.

Update also checks the current error count and halts the programme if this exceeds a preset limit (these errors are mainly concerned with i/o or file failures and are counted, and the programme allowed to continue past the error, only if the feature NOERROR has been enabled in the top-level file).

I/O:

Identifiction to default device if required.

ERROR TRAPPING:

Programme execution is halted if model error count exceed preset maximum.

CHAPTER 6

RADIANT TRANSFER

These routines deal with the exchange of radiant energy between the internal surfaces of the building system, and with the absorption of energy from irradiation of internal and solar sources.

TEMPRD - star temperature radiant transfer . . 6-2
MRTCLC - calc mean radiant temperature . . . 6-7
EXCTRD - matrix equation radiant exchange . . 6-11

6.1 TEMPRD - star temperature radiant transfer - level 1

CALLED FROM : HTBOS

CALL FORMAT : CALL TEMPRD

CALLS TO : MRTCLC

CALL FREQUENCY : every timestep, use conditional on choice of calculation procedure.

GENERAL DESCRIPTION :

TEMPRD applies the radiant output calculated from each process (ie. plant,incidentals,solar) to the surfaces of a space, and the determines the intersurface longwave transfer, to calculate the total radiant heat absorbed at each surface. An equivalent star network approximation is used. The effective blackbody temperature seen by each surface is also calculated.

The use of TEMPRD is conditional on the option variable LCLCRD. This routine is used if LCLCRD = 1 (chosen in the input stage by !ENABLE RADTRAN STAR).

THEORETICAL BASIS:

Each appropriate transfer process, ie. heating plant, incidentals, solar will have produced as a reaction to the conditions of the last timestep, a radiative power. Depending on the source, this power may be directed to nominated surfaces, in W/m2, or it may be considered as diffuse gains to a radaint zone, in watts. These powers are at this stage accumulated. Direct irradiance is applied to surfaces, weighted by surface emissivity. The total reflected power from the direct sources, and the diffuse gains in a zone are apportioned to all surfaces in that zone by a surface area-emissivity weighting. As each process is applied it is reset to zero for the next round of calculations, except for the solar irradiance which is held constant until recalculated by the solar routines.

The radiant transfers are organised on the basis of zones rather than spaces. In general a zone will be identical to a space, however a zone can be defined to encompass more than one space, thus allowing radiant transfer across space boundaries (which may well be imaginary or virtual).

Radiant gains from heating system, small power sources, insolation, and diffuse sources are considered to be from high temperature sources, so that shortwave emissivity (or surface absorption) is used for these.

Internal longwave transfer between surfaces is calculated by determining an equivalent star network. The star temperature is determined from the area-longwave emissivity weighted radiosities of the surfaces. The net gain of each surface is calculated relative to this star point.

①A mean surface temperature, and ②an effective mean radiant temperature, for each zone, is calculated in the routine MRTCLC.

Effective radiant temperatures "seen" by each surface are calculated from the net radiative gain from both surface and incidental sources. The effective radiant temperature is defined to be the equivalent blackbody required to produce the same net radiant transfer.

Finally the net radiant gain to external surfaces are calculated from the (predetermined) solar irradiance, longwave gain from the sky vault, and the surface temperature.

At first call, or on a flagged change of surface characteristics, the weighting variables are calculated for later use.

The calculations used in the procedure are;

The total radiant gain (W/m2) to an internal surface (i) is

$$\boxed{Rt_i} = \boxed{Rs_i} + \boxed{Rl_i}$$

where the irradiance from local sources, Rs , is determined from

$$\boxed{Rs_i} = \overset{\text{direct}}{\boxed{es_i * Idr_i}} + \overset{\text{weighted reflected}}{\boxed{(\sum_j (1- es_j)* Idr_j * A_j)* Aes_i}} + \overset{\text{Weighted diffuse}}{\boxed{Idf_i * Aes_i}}$$

where Aes is the short-wave area-emissivity weighting for the surface i, given by

$$Aes_i = \frac{A_i * es_i}{\sum_j (A_j * es_j)} \text{, over all surfaces j in the zone.}$$

The intersurface radiant exchange, Rl , is determined from

$$\boxed{Rl_i} = \sigma * el_i (Ts_i^4 - Tn^4)$$

where Tn is the equivalent "star" temperature (oK), calculated as

$$Tn = (\sum_i Ts_i * Ael_i)$$

where Ael_i is the long-wave area-emissivity weighting for the surface i, given by

$$Ael_i = \frac{A_i * el_i}{\sum_j (A_j * el_j)}$$, over all surfaces j in the zone.

The net radiant gain for an external surface ,Rt_i (W/m2), is determined from,

$$Rt_i = es_i * Islr_i + (Ilng_i - el_i * \sigma * Ts_i^4).$$

The effective radiant temperature ,Tsn (oK) as 'seen' by a surface is determined as

$$Tsn_i = \left(\frac{Rt_i}{\sigma * el_i} - Ts_i^4 \right)^{1/4} .$$

For all these equations

es	is the shortwave emissivity of the surface,
el	is the longwave emissivity of the surface,
A	is the surface area (m2),
Ts	is the surface temperature (oK),
σ	is the stefan-boltzman constant,
Idr	is the direct irradiance for internal surfaces (W/m2),
Idf	is the diffuse irradiance in an internal zone (W),
Islr	is the net solar irradiance on external surface (W/m2),
Ilng	is the net longwave irradiance on external surface (W/m2).

I/O:

Identification to default device if required.

6.2 MRTCLC - calc mean radiant temeprature - level 2

CALLED FROM : TEMPRD

CALL FORMAT : CALL MRTCLC(IZN)
: IZN - intg - current zone number

CALLS TO : -

CALL FREQUENCY : for every zone, for every call to TEMPRD

GENERAL DESCRIPTION:

MRTCLC calculates a mean surface temperature, and a mean radiant temperature, for the zone given as the arguement.

THEORETICAL BASIS:

The mean surface temperature Tur (oC), is calculated as an area weighted mean,

$$Tur = \sum_{j} Ts_{j} * \frac{A_{j}}{At_{i}}$$

where A is the surface area (m2),
At is the total surface area in zone (m2)
Ts is the surface temperature (oC),
over all surfaces j of the zonei.

An effective mean radiant temperature, Tmr (oC), including the net radiant energy from sources in the zone, is calculated as

$$Tmr_{i} = Tur_{i} + \frac{Rt_{i}}{At_{i}} * \frac{1}{Hr}$$

where Rt is the net radiant energy in the zone (W), determined from the net diffuse energy and the total direct surface irradiation in the zone,
Hr is a linearised radiant transfer coefficient (W/m2/oC), a parameter determined from

$$Hr = 4 * \sigma (Tr)^{3}$$

where is the stepan-boltzman constant
Tr is a mean reference temperature (oK), taken from the model reference temperature TREF, as defined in the top-level file.

I/O:

6.3 EXCTRD - matrix equation radiant exchange - level 1

CALLED FROM : HTBOS

CALL FORMAT : CALL EXCTRD

CALLS TO : -

CALL FREQUENCY : every timestep, use conditional on choice of calculation procedure.

GENERAL DESCRIPTION:

EXTCRD is a dummy routine. A matrix solution of internal longwave radiant transfer, incorporating viewfactors, is under development.

Calling this dummy routine will cause an error condition and halt the programme.

I/O:

Identification to defualt device if requested.
Error and warning messages to default device.

ERROR TRAPPING:

Calling dummy routine causes error and aborts programme.

CHAPTER 7

METEOROLOGICAL DATABASE

These routines deal with the updating of the meteorological database from information supplied in an external file. Required information is external temperature, wind speed and direction, and total horizontal irradiance. These routines provide values for sun position, direct and diffuse horizontal irradiance, direct normal irradiance, external surface irradiance, longwave gain and convective transfer coefficients.

METBAS - read and maintain met database 7-2
  EXTRHT - determine external convection coeffs . 7-7
  LNGWAV - determine external longwave loss . . . 7-11

7.1 METBAS - read and maintain met database - level 1

CALLED FROM : HTBOS

CALL FORMAT : CALL METBAS

CALLS TO : CRASHR , EXTRHT , FORERR , LNGWAV , RH2DP , RH2VP , RH2WC

CALL FREQUENCY : at defined meteorological interval

GENERAL DESCRIPTION:

METBAS determines the external conditions to be used over the meteorological interval. Data is read from an external file defining the external temperatures, wind and solar conditions. Solar irradiance, surface transfer coefficients, and longwave losses are calculated.

THEORETICAL BASIS:

Data is read from an external file in list directed format. Each record, for each meeorological interval, is read as (in list directed format);

DUMMY , EXTT , WVEL , WDIR , RH , CC , DIF , TOT , GRND

where DUMMY is a number for record identification , not used in HTB2,
EXTT is the external air temperature (C),
WVEL is the wind speed (m/s),
WDIR is the wind direction (o east of north),
RH is the external air humidity (%),
CC is the total cloud cover (0 - 1),
DIF is the diffuse orizontal irradiance (W/m2),
TOT is the total horizontal irradiance (W/m2),
GRND is the deep ground temperature (C).

DUMMY is expected to be integer, while all others are real.

From this data the following parameters are calculated; direct horizontal irradiance, direct normal irradiance, external air water content, solar declination, altitude, and azimuth, angle of solar incidence for each external surface, external surface convective heat transfer coefficients, and external surface longwave radiant exchange.

Direct horizontal irradiance is determined from the subtraction of diffuse horizontal from the total horizontal irradiance, while direct normal is determined from

$$Idn = \frac{Ith}{SIN(\beta)}$$

where β is the solar ltitude.

Solar geometry is determined from standard algorithms.

Given lat the site latitude,
long the site longitude,
lngs the longitude of the local time zone,
W the day angle, (all degrees)
and H the local time (hours).

solar azimuth γ ;

$$\gamma = ATAN \frac{SIN(t)}{COS(t) * SIN(lat) - COS(d) * COS(lat)}$$

where t is the solar hour angle as given by

$$t = 15 * (H + ET - 12) + (Long - Lngs),$$

where ET is given by the equation of time,

$$ET = 0.007 * COS(W) - 0.05 * COS(2W) - 0.0015 * COS(3W) - 0.122 * SIN(W) - 0.150 * SIN(2W) - 0.005 * SIN(3W),$$

and d is the solar declination as determined from

$$d = 0.302 - 22.93 * COS(W) - 0.229 * COS(2W) - 0.243 * COS(3W) + 3.851 * SIN(W) + 0.002 * SIN(2W) - 0.055 * SIN(3W) .$$

solar altitude β ;

$$\beta = ASIN \{ SIN(lat) * SIN(d) + COS(lat) * COS(d) * COS(t) \}$$

External convective heat transfer coefficients, and external surface longwave irradiance, are determined for the current meteorological interval in the routines EXTRHT and LNGWAV.

Where calculation has been selected in top-level file, the ground temperature, Tg (oC), specified in the data is overwritten by a temperature determined from the external air temperature, Tair, as

$$Tg = F * (Tair - Tg')$$

where F is a factor specified in the top-level file, and

Tg' is the previous interval ground temperature.
This corresponds to an exponential attenuation of air temperature with a time constant (hour) of

$$\tau = \frac{Im}{F}$$

where Im is the meteorological data interval in hours.

I/O:

Identification to default device if required.
Sequential, formatted, list directed input from external file.

ERROR TRAPPING:

Fatal trap at end of file on read.

7.2 EXTRHT - determine external convection coeffs. - level 2

CALLED FROM : METBAS

CALL FORMAT : EXTRHT(K,T,W) - function call
: K - intg - element pointer
: T - real - element tilt
: W - real - element azimuth

CALLS TO : -

CALL FREQUENCY : for every external element, for every call to METBAS

GENERAL DESCRIPTION :

EXTRHT determines the convective heat transfer coefficients at external surfaces to be used over the meteorological interval. These coefficients are precalculated here for later use be the fabric transfer routine. The algorithm is taken from HTB.

THEORETICAL BASIS :

The external convective transfer coefficients for each external surface , He (W/m2/oC), are determined from the windspeed and the element orientation relative to the wind direction.
There are held to be three cases of orientation;

i - horizontal element -

$$He = 18.624 * (.25 * U)^{.605}$$

if wind speed > 2.0 m/s, or

$$He = 12.25$$

if wind speed < 2.0 m/s.

ii - vertical windward -

$$He = 18.624 * (.25 * U)^{.605}$$

if wind speed > 2.0 m/s, or

$$He = 12.25$$

if wind speed < 2.0 m/s.

iii - vertical leeward -

$$He = 18.624 * (0.3 + 0.05 * U)^{.605}$$

where U is the site wind speed (m/s).

This formulation gives a good approximation to published data, i.e. fig 3.9 of KIMURA,K "Scientific Basis of Air Conditioning", Applied Science Publishers Ltd., London, 1977. with the possibility of over-estimation at low wind speeds.

I/O:

Identification to default device if required.

7.3 LNGWAV - determine external longwave gain - level 2

CALLED FROM : METBAS

CALL FORMAT : LNGWAV(K,T,O) - function call
: K - intg - element pointer
: T - real - element tilt angle
: O - real - element orientation

CALLS TO : -

CALL FREQUENCY : for every external element, for every call to METBAS

GENERAL DESCRIPTION:

LNGWAV determines the longwave radiation incident to external surfaces, from the environment. The algorithm is taken from HTB. The net longwave exchange is determined in the routine TEMPRD.

THEORETICAL BASIS

The longwave irradiation ,Ilng (W/m2), is determined for each external surface from the external air temperature, cloud cover, and element orientation as follows;

Given Tair , the external air temperature (oC),
cc , the cloud cover (tenths),

the horizontal irradiance from the sky vault, Ih (W/m2), is determined from

$$Ih = 222 + (4.94 * Tair) * (8 * CC),$$

irradiance on a vertical surface from the sky vault ,Iv (W/m2), is given by

$$Iv = 0.5 * Ih + 0.3457 * Bb * \sigma * (Tair+273),$$

and irradiance on a vertical surface from the ground, Ig (W/m2), is given by

$$Ig = 162 + 3.15 * Tair - (8 * CC * Bb).$$

Bb is determined from

$$Bb = 0.09 * (1 - CC * (0.7067 + 0.00822 * Tair)).$$

Thus for horiontal surfaces,

Ilng = Ih

while for vertical surfaces,

Ilng = Iv + Ig .

These algorithms are simpified forms of the equations reported in
AUSTIN,M. , COLE,R.J. " A THERMAL DESIGN MODEL BASED ON ENERGY EXCHANGE: Part2 The prediction of the atmospheric radiation incident at the external surface of buildings." , 1974
SERC ref B/RG/2472

I/O:

Identification to default device if required.

CHAPTER 8

INSOLATION

These routines deal with the determination of insolation and solar irradiance to internal surfaces.

- SOLAR - determine surface irradiance from solar . . 8-2
  - SKYVW - determine element obstruction 8-6
  - TRFNCT - detemine transmission of window 8-11

8.1 SOLAR - determine surface irradiance from solar - level 1

CALLED FROM : HTBOS

CALL FORMAT : CALL SOLAR

CALLS TO : SKYVW , TRFNCT

CALL FREQUENCY : every meterological interval, or when directed due to a change in building or shading description.

GENERAL DESCRIPTION:

SOLAR manages the calculations of shaded irradiance to external surfaces, window transmission and absorption, and irradiance to internal surfaces. The calculations are done when new meteorological information is read, and are held constant between calculations. Solar recalculation can also be forced at any time, ie if a transmission characteristic is altered, by the flag LFRCSL.

THEORETICAL BASIS:

Data maintained in the meteorological database by METBAS include diffuse horizontal irradiance, direct normal irradiance, and sun position. These, with the description of surface orientation and sky shading masks provided in the input stage, are used to calculate direct and diffuse solar irradiance to external surfaces.

The surface orientation, ω ,and sun azimuth, γ (both degree c.w from south), are combined to produce a relative azimuth, Rz , by

$$Rz = \gamma - \omega \quad .$$

The incidence angle to the surface is determined from

$$Inc = ACOS (COS(\beta) * COS(Rz) * SIN(\phi) + SIN(\beta) * COS(\phi)) \quad ,$$

where ϕ is the surface tilt (degrees from horizontal to surface normal,
β is the solar altitude.

The total diffuse solar irradiance, Idf (W/m2), on the surface i is calculated from

$$Idf = SFdf_i * Idh + Ith * Gr_i * (1 - COS(90-\phi)) / 2 \quad ,$$

where SFdf is the diffuse sky shading factor, as provided by SKYVW,
Idh is the diffuse horizontal irradiance (W/m2),
Ith is the total horizontal irradiance (W/m2),
Gr is the reflectance of adjacent ground,
ϕ is the surface tilt.

The direct solar irradiance, Idr (W/m2), on an external surface is determined from

$$Idr = SFdr * Idn * COS(Inc),$$

where SFdr is the direct obstruction factor, as provided by routine SKYVW,
Idn is the (unobstructed) direct normal irradiance (W/m2),
Inc is the incidence angle to surface.

Where an element is opaque, these direct and diffuse components are simply combined to give the net shortwave solar irradiance to the surface.

Transparent elements will also have had specified, transmission and absorption characteristics according to incident angle. These are used to determine the insolation to the building interior and the internal absorption to the fabric. Diffuse and direct ransmissions are treated separately. Diffuse transmission is determined in watt gain to a radiant zone, and this total is apportioned to zone surfaces in TEMPRD. Direct beam transmission is apportioned to surfaces according to the sunpatching specification for the transmitting element provided at the input stage. This 'sunpatching' may, for instance, simply be an instruction to place solar gains on the floor surface. There is no provision, at present, for a dynamic sunpatch calculation.

I/O:

Identification to default device if required.

8.2 SKYVW - determine element obsruction - level 2

```
CALLED FROM       : SOLAR

CALL FORMAT       : SKYVW(K,ALT,AZI,I)  ( function call)
                       : K - intg - element pointer
                       : ALT - real - solar altitude
                       : AZI - real - solar azimuth
                       : I - intg - function code = 0 - diffuse
                                                  = 1 - direct

CALLS TO          : CRASHR

CALL FREQUENCY    : for each element, for each call to SOLAR
```

GENERAL DESCRIPTION:

SKYVW determines diffuse and direct obstruction (or attenuation) of solar energy falling on an external surface, according to a predefined sector mask for the surface, and the solar position.

THEORETICAL BASIS:

Each external element will have had a sector mask nominated for it either through the initial input stage or the diary. Masks are defined in the input stage and accessed by a character name. There is one default mask, 'NONE', which corresponds to an unobstructed site.

A mask is defined as the transmission factor of the sky vault in 10 degree sectors. Such a mask may be determined from a stereographic projection of site obstructions as shown in figure 8.1. They are always specified from a horizontal plane with the central point placed wherever appropriate on the surface. Azimuth angles are determined clockwise from south.

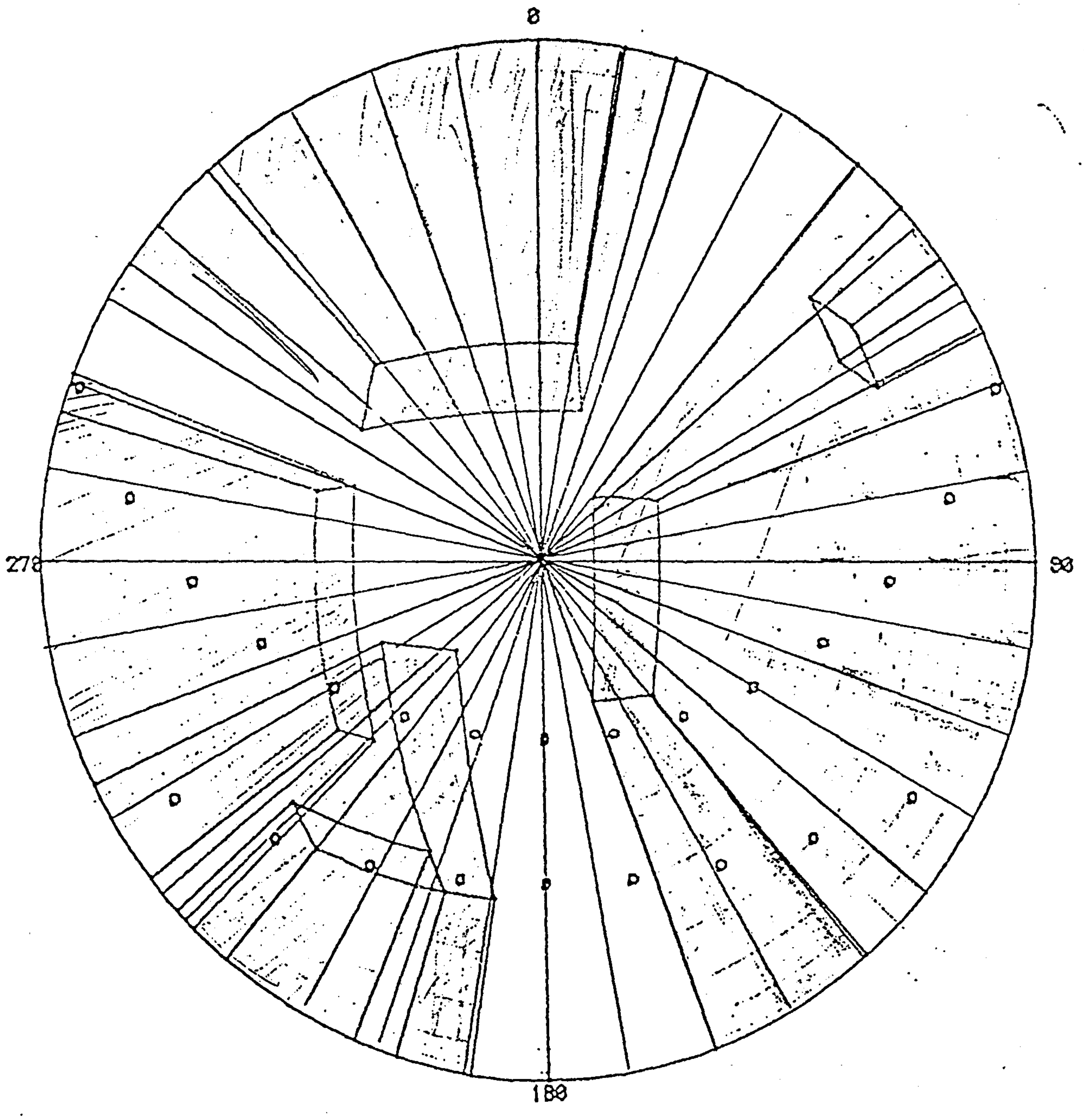


fig 8.1 SKY OBSTRUCTION

The diffuse sky vault is considered to be isotropic so that transmission of the horizontal diffuse irradiance is determined from the mean sector transmission for defined masks. If the default ('NONE') mask has been chosen the diffuse transmission factor, SFdf , is determined from the surface tilt, ϕ , by

$$SFdf = (1 + COS(90 - \phi))/2.$$

The transmission factor of the direct beam is determined according to the specified transmission of the sector currently occupied by the sun. In the case of the default mask this transmission is always 1.0 .

I/O:

Identification to default device if required

ERROR TRAPPING:

Halts programme if undefined masking template required.

8.3 TRFNCT - determine transmission of window - level 2

CALLED FROM : SOLAR

CALL FORMAT : TRFNCT(K,AINC,I,J) (function call)
: K - intg - element pointer
: AINC - real - incidence angle
: I - intg - function 1 code = 0 transmission
1 absorption
: J - intg - function 2 code = 0 diffuse
1 direct

CALLS TO : CRASHR

CALL FREQUENCY : for every transparent element, for each call to SOLAR

GENERAL DESCRIPTION:

TRFNCT provides the transmission and absorption factor of incident solar energy, for both direct and diffuse irradiance, for transparent elements.

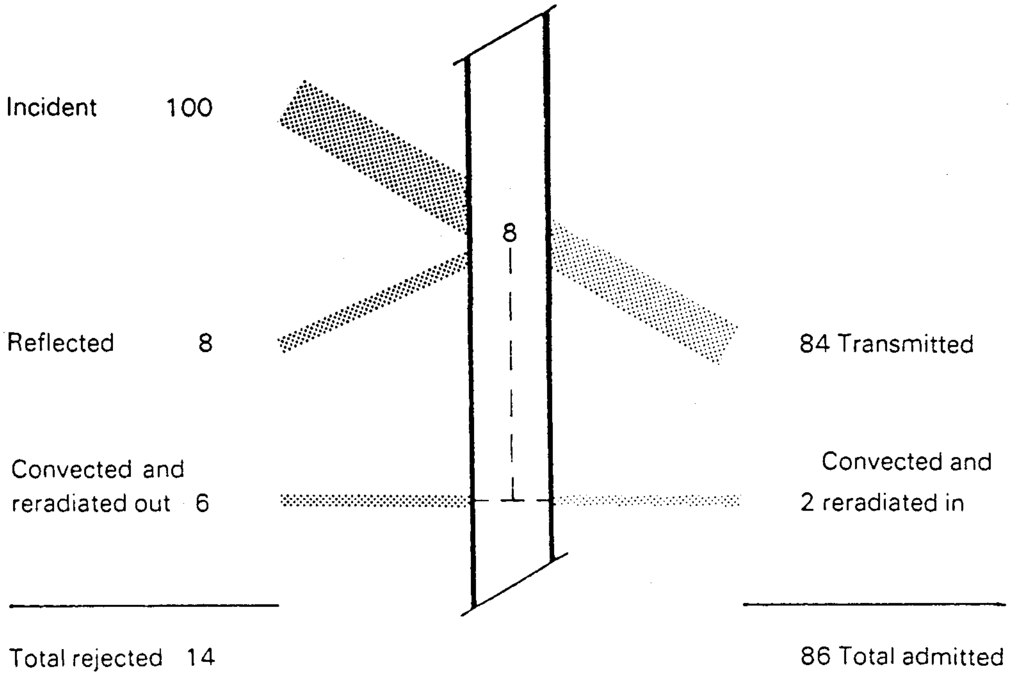
THEORETICAL BASIS:

Each transparent element will have had a transmission characteristic nominated for it either through the initial input stage or the diary. The characteristics are defined in the input stage and accessed by a character name.

A characteristic is defined as the total transmission, and total absorption factors of incident energy, for incident angles in 10 degree steps for direct beam, and global factors for diffuse irradiance. Such a characteristic may be determined from manufacturers data, or theoretical calculation. A typical characteristic for 4mm clear float glass is shown in figure 8.2.

The transmission or absorbtion characteristic determined for a particular incidence angle is determined from interpolation on the specified data.

4mm clear sheet



Data for near normal incidence

Solar transmittance	T	0·84
Solar absorptance	A	0·08
Solar reflectance	R	0·08
Partition factor	P	0·31
Total heat gain factor	F	0·86
Shading coefficient	S	0·99
Light transmittance		0·88

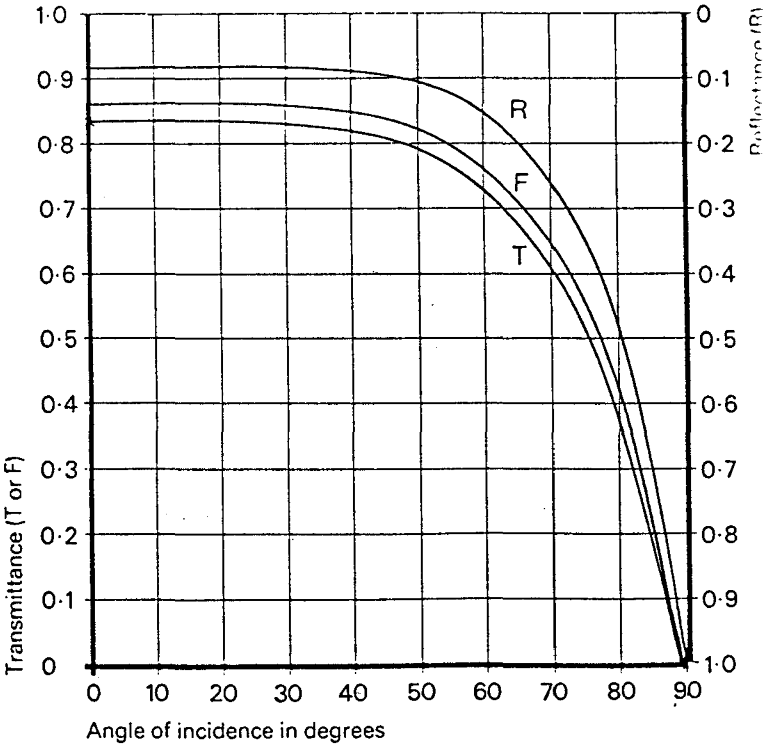


Fig 8.2 4mm clear Float glass (cf. PILKINGTON DATA SHEETS)

I/O:

Identification to default device if requested.
Error and warning messages to default device.

ERROR TRAPPING:

Halts programme execution if unknown transmission characteristic code required.

CHAPTER 9

FABRIC HEAT TRANSFER

These routines deal with the calculation of fabric heat transport and fabric temperatures.

FABRIC - calculate fabric transfer 9-2
  CAVTRN - determine cavity transfer 9-9
  HTCE - chose external convective coefficient . 9-14
  HTCF - chose internal convective coefficient . 9-17
  HTCX - chose air-air convective coefficient . . 9-20

9.1 FABRIC - calculate fabric transfer - level 1

CALLED FROM : HTBOS

CALL FORMAT : CALL FABRIC

CALLS TO : CAVTRN , HTCE , HTCF , HTCX

CALL FREQUENCY : every time-step

GENERAL DESCRIPTION:

FABRIC determines the heat flows through the building fabric, calculating new fabric temperatures due to heat gains from air, radiant, and direct conduction sources. Radiant gains to surfaces, and direct gains to elements have already been established at this point. Convective heat transfer is calculated here and the net heat transfer from space air is determined.

THEORETICAL BASIS:

An explicit finite difference procedure is used to calculate the heat flow through, and new temperatures of, the fabric. The general form of this is,

$$_{k+1}\Theta_m = P\left[{}_k\Theta_{m+1} + {}_k\Theta_{m-1} + \left(2 - \frac{1}{P}\right){}_k\Theta_m\right]$$

where
k, m denote time and space increments,
$P = A\,\Delta T/\Delta x^2$
A is the diffusivity of the material,

Note that the calculations depend only on the existing (known) temperatures to determne the heat flows and new temperatures, and that for mathematical stability the term in present time, for present slice, must be positive, giving rise to the stability criterion mentioned in the discussion of DEFBLD.

In practice each element (a wall, ceiling etc.) is described by layers of materials, each with specified properties of condctivity, density, specific heat and width. Each layer is divided into discrete 'slices' of equal thickness (within a material) and the temperature of each slice is approximated by calculating the temperature at the center of the slice. As well as the surface temperatures and the net heat transport, this method determines the temperature and heat flux profile of each element. Temperatures are stored for the centre of each slice, for boundaries between materials, and for surfaces, while heat fluxes are stored for the boundaries between slices, materials and surfaces.
The general procedure used for each slice of an element is;
i determine heat flux into slice from boundary conditions,
ii determine heat flux out of slice from adjacent slice

temperature,
iii calculate new slice temperature from net heat gain to slice,
iv t carry heat flux out of slice to heat flux into next slice as a boundary condition,
repe~~a~~ted for the next slice in the chain.

The particular procedure used depends in effect on the position of the slice. The classes of slices are;

i slice entirely within a material,
ii slice at a boundary of two materials,
iii slice at a boundary of material and internal space,
iv slice at a boundary of material and the exterior.
v slice at a boundary of material and a cavity,

Calculations of heat flux into a slice are made using ;

i For a slice within a material, ie the same material in the previous slice, the heat flow is taken as unobstructed,

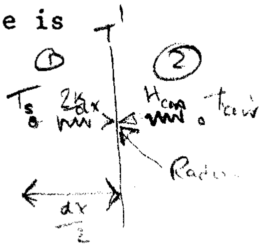
$$Hin = Hout,$$

where Hout is the previously calculated heat flux out of the preceeding slice.

ii For a slice at a boundary with another material, ie with a different material proceding, the same assumption is used,

$$Hin = Hout.$$

iii For a slice at a boundary with an internal space, ie the slice has a surface boundary preceding, the surface temperature is unknown and must be solved.
In general,



$$Hin = 2k/dX\ (T' - Ts), \qquad ①$$

where k is the thermal conductivity of the slice,
T' is the unknown surface temperature,
Ts is the known slice temperature,
dX is the slice thickness.

Also the heat flux in will be the heat flux through the surface boundary, so that

$$Hin = Hcon\ (Tair - T') + Radin; \qquad ②$$

where Hcon is the convective heat transfer coefficient at the surface,
Tair is the space air temperature, and
Radin is the net radiant exchange at the surface (calculated in routine TEMPRD or EXCTRD).

Hence these equations may be combined to solve for the surface temperature T',

① = ②

heat
Zero/gain at surface

$$T' = \frac{2ks/dXs\ Ts + Hcon\ Tair + Radin}{2ks/dXs + Hcon},$$

and that used in the general equation to determine the heat flux,

$$Hin = 2k/dX\ (T' - Ts).$$

The value for Hcon is determined in routine HTCF, which chooses from three specified values according to the surface orientation.

iv For a slice at a boundary with the exterior, ie the slice has a external surface boundary preceding, the approach is similar to case iii above, with a value of Hcon appropriate for the exposure of the external surface, determined in routine HTCE, and Radin incorporating solar gains and longwave losses (both determined from the met data).

v For a slice at a boundary with a cavity a separate routine CAVTRN determines the surface temperature and heat flux.

Heat flux out of a slice is calculated from;

vi For a slice within a material, ie the same material in the next slice, the heat flow is determined from the two slice temperatures as

$$Hout = k/dX\ (Ts - Tn),$$

where k is the thermal conductivity of material,
dX is the thickness,
Ts is the slice temperature, and
Tn is the adjacent slice temperature.

vii For a slice at a boundary with another material, ie with different material following ,there exists an unknown interface temperature between the materials which depends on the temperatures and properites of the two materials. The general equations for heat flow out of the slice to next, and for heat flow into the next from the slice, using an intermediate boundary temperature are,

$$Hout = 2ks/dXs\ (Ts - T'),$$

and

$$Hin = 2kn/dxn\ (T' - Tn),$$

where ks,kn are the conductivities of the slice and next material respecively,
dXs,dXn are the thicknesses of the two slices,
Ts,Tn are the slice temperatures, and
T' is the interface temperature.

Solving for T' this gives,

$$T' = \frac{2ks/dXs\ Ts + 2kn/dXn\ Tn}{2ks/dXs + 2kn/dXn},$$

and

$$Hout = 2ks/dXs\ (Ts - T').$$

viii For a slice at a boundary with an internal space, ie the slice has a surface boundary following, a similar procedure to determine the surface temperature is needed, replacing the adjacent slice temperature and resistance with air temperature, convective transfer coefficient, and net radiant transfer as used in case iii for heat flow in.

$$T' = \frac{2ks/dXs\ Ts + Hcon\ Tair + Radin}{2ks/dXs + Hcon},$$

and $$Hout = 2ks/dXs\ (Ts - T'),$$

where ks,dXs and Ts are defined as in case vii above and Tair,Hcon and Radin are defined as in case iii.

The value Hout is accumulated for each internal, modelled, space to determine the net heat transfer from space air. This is used in routine TEMPSP in conjunction with net convective gains from other processes to determine the new space temperature.

ix A slice at a boundary with the exterior, ie the slice has a external surface boundary following, is only met in the case of a ground floor (the description of the element structure is defined from the exterior surface to the interior surface, except in the case of a ground floor). Here the interface temperature is taken to be some defined value for 'ground' temperature (which may also represent a subfloor void temperatue) and

$$Hout = 2ks/dXs\ (Ts - Tbelow),$$

where Tbelow represents a 'ground' temperature and is determined from the meteorological data.

Cavities are treated separately in the routine CAVTRN. As the cavity is not treated explicitly in the finite difference calculations, this routine is responsable for determining the surface and cavity temperatures, and the heat flux out of the preceding slice and into the following slice. This routine may incorporate cavity losses due to ventilation.

After the heat flux in and out of a slice has been determined a new slice temperature is calculated from the net heat gain to the slice,

$$Ts' = Ts + (Hin - Hout + Hdir) * A * Capac,$$

where

Ts, Ts'	are the old and new slice temperatures respectively,
Hin,Hout	are the heat flows in and out as calculated above,
Hdir	represents any internal gains to the slice,
? A	is a conversion factor from heat flux (w/m2) to energy (j), and
Capac	is the heat capacity of the slice material.

A further, special, boundary may be specified which may act to separate two air spaces. This is termed a 'virtual' partition, in that it does not represent a physical boundary, and it may be useful in simulating stratification or zoning of large spaces. In this case the heat transfer across the boundary is a simple function of the two adjacent air temperatures and anair to air heat transfr coefficient.

$$H = Htc * (Tair1 - Tair2),$$

where

H	is the heat transfer rate between the two spaces,
Htc	is an air to air heat transfer coefficient,
Tair1,Tair2	are the air temperatures of the two spaces respectively.

The net heat transfer of the two spaces are accumulated as in the case viii above.

The usefulness of such a boundary will depend on the choice of the air-air heat transfer coefficient. At the moment the choice is made in routine HTCX, which simply chooses from three specified coefficients according to orientation.

The heat transferred from internal air through internal surfaces is accumulated to be applied later to determine the new air temperature (TEMPSP).

I/O:
Identification to default device if required.

9.2 CAVTRN - calculate cavity transfer - level 2

CALLED FROM : FABRIC

CALL FORMAT : CALL CAVTRN(N,P,K,M,H1,H2)

N > data node pointer , integer
P > construction part pointer , integer
K > element pointer , integer
M > cavity type , integer
H1 < heat flux from slice to cavity, real
H2 < heat flux from cavity to next slice, real

CALLS TO : -

CALL FREQUENCY : every timestep for every cavity

GENERAL DESCRIPTION:

CAVTRN determines the temperature and heatflux through a cavity located by N , P , K. It also sets up the surface temperatures and the heat flux out of the preceding slice, H1 , and the heat flux out of the cavity into the following slice, H2 so as to tie into the calculation procedure of the main fabric trnsfer routine FABRIC. This version treats only unventilated cavities, however there may be an internal source of heat, such as absorbed solar energy.

THEORETICAL BASIS:

A cavity is considered to be a simple extremely lightweight material (i.e. steady state approximations are made and thermal capacity is ignored), so that it is treated only as an extra resistance between two adjacent materials. A cavity is restricted to only one slice. The effective electrical analogue is shown in fig 9.1. The cavity temperatures, and boundary surface temperatures and heat fluxes are to be determined from the known fabric temperatures and the known internal heat gain.

The cavity transfer resistance is determined according to the cavity type; high or low emissivity, and the thickness. Standard handbook values are used.

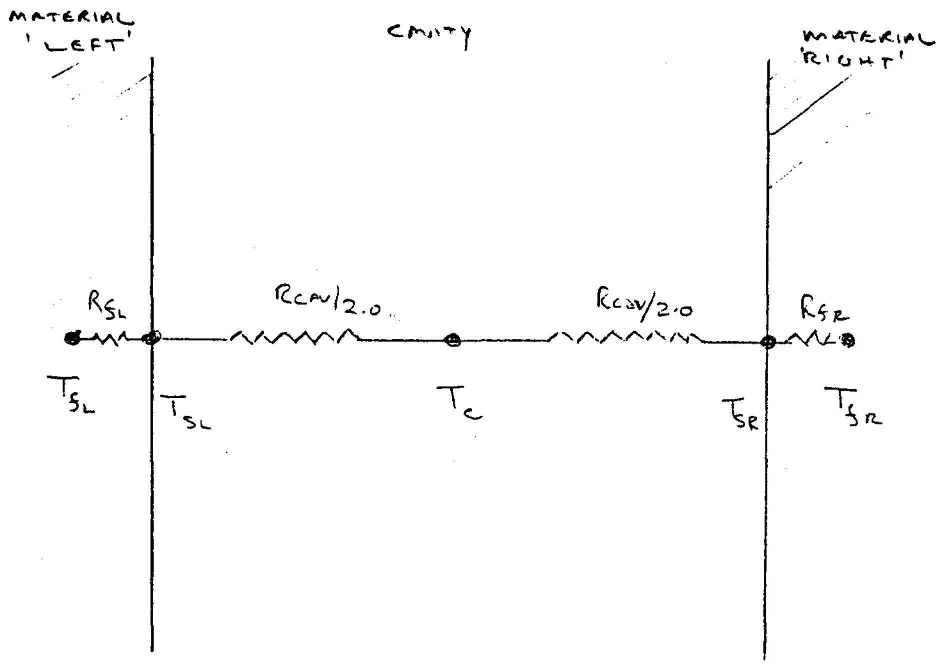


Fig 9.1 CAVITY ELECTRICAL ANALOGUE

Given the bounding fabric temperatures, Tf1 , Tf2 (oC),
the bounding fabric conductivities k1 , k2 (W/m/oC),
the bounding fabric slice thickness X1 , X2 (m)
the cavity resistance , Rc (oCm2/W),
and the cavity internal heat gain , Hc (W/m2), then

The fabric resistances Rf1, Rf2 (oCm2/W), are determined from

$$Rf1 = X1/k1 \quad , \; Rf2 = X2/k2$$

The new cavity temperature Tc is given by,

$$Tc = \frac{Tf1*(Rf2+Rc/2) + Tf2*(Rf1+Rc/2) + Hc*(Rf1+Rc/2)*(Rf2+Rc/2)}{Rf1 + Rf2 + Rc}$$

and so the heat flux from the left bounding surface, H1 (W/m2), is

$$H1 = \frac{Tf1 - Tc}{Rf1 + Rc/2}$$

with a similar calcualtion for the right surface heat flux. The left surface temperature, Ts1 (oC), is then determined by

$$Ts1 = Tf1 - (Rf1 * Rc/2) * H1 \; ,$$

with a similar calculation for the right surface temperature.

The surface heat fluxes are returned in variables HFOUT and HFOUTC respectively to interface with the finite difference procedure of FABRIC.

I/O:

Identification to default device if required.

9.3 HTCE - chose external convective coefficient - level 2

CALLED FROM : FABRIC

CALL FORMAT : HTCE(N,K,P,I) - function call
N > data node pointer , integer
K > element pointer , integer
P > construction part pointer , integer
I > surface pointer, integer
(these arguments are for future use)

CALLS TO : -

CALL FREQUENCY : every timestep, for every external surface

GENERAL DESCRIPTION:

The function HTCE returns a value for the convective heat tranfer coefficient at the external surface described by N , K , P, I .

THEORETICAL BASIS:

External convective heat transfer coefficients have been calculated and stored for each element with an external face, by the meterological routines METBAS and EXTRHT. This routine simply retrieves the appropriate value.

I/O:

Identification to default device if required.

9.4 HTCF - chose internal convective coefficient - level 2

CALLED FROM : FABRIC

```
CALL FORMAT   : HTCF(N,K,P,I,S) - function call
                     N > data node pointer , integer
                     K > element pointer , integer
                     P > construction part pointer , integer
                     I > surface pointer , integer
                     S > connecting space pointer , integer
```

CALLS TO : -

CALL FREQUENCY : every timestep, for every internal surface

GENERAL DESCRIPTION :

The function HTCF returns as its value, a convective heat transfer coefficient appropriate for the internal surface located via the arguments.

THEORETICAL BASIS:

This routine chooses a coefficient for an internal surface from three predefined values for heat flow horizontal, heat flow upwards and heat flow downwards. The orientation of the element (the tilt angle) and the direction of the temperature gradient between the surface and the adjacent space air are used to differentiate between the three conditions. The coefficients are defined as parameters in the main input sequence, and are standard textbook values by default.

I/O:

identification to default device

9.5 HTCX - chose air-air convective coefficient - level 2

CALLED FROM : FABRIC

CALL FORMAT : HTCX(T1,T2,K) - function call
T1 > temperature on LHS , real
T2 > temperature on RHS , real
K > element pointer , integer

CALLS TO : -

CALL FREQUENCY : every timestep for every virtual partition

GENERAL DESCRIPTION:

The function HTCX returns a value for air to air convective heat transfer coefficients appropriate to the virtual element KEL and the adjacent air temperatures.

THEORETICAL BASIS:

This routine selects one of three predefined coefficients according to the direction of the heat flow expected; horizontal, upwards, downwards. The argument T1 is defined as the temperature to the left of the virtual element and the specified tilt of the element is used to determine the flow direction. The three values are specified as parameters in the main input sequence, a separate set of values may be specified for each element. There is no implication that the current defaults for these parameters of 10 W/m2/oC have any validity. Considerable work is needed on the movement of air within a space before such a simplification can be reasonably made. This feature of HTB2 is included in anticipation of such data.

I/O:

Identification to default device if required.

CHAPTER 10

VENTILATION TRANSFER

These routines deal with the transport of heat and water vapour by air movement.

- VENTL - calculate ventilation transfer 10-2
  - VENT1 - chose space air change rates 10-6
  - VENT2 - chose space to space flow pattern . . 10-10
  - VENT3 - calculate empirical air change rate . 10-13
  - VENT8 - use multicell mathematical model . . 10-17

10.1 VENTL - calculate ventilation transfer - level 1

CALLED FROM : HTBOS

CALL FORMAT : CALL VENTL

CALLS TO : VENT1 , VENT2 , VENT3 , VENT8

CALL FREQUENCY : every time-step

GENERAL DESCRIPTION:

VENTL determines the net heat and water movement between spaces from the existing airflow rates, and calculates the total and fresh (external air entry) air change rates. It also manages the routine which determines the airflow rate and pattern appropriate to the time and conditions.

THEORETICAL BASIS:

A connection matrix exists which specifies the air movement between each space and all other spaces (including the exterior and the 'virtual' spaces), in m3/s, for both air entry and exit. It is this array of intercell air movement that is used to calculate the net heat and water exchange between spaces (and between spaces and the exterior). The values in the array are determined in the calls to the appropriate ventialtion calculation routines (i.e. VENT1,VENT2).

The net heat change in space air, due to ventilation transfer, Hv (W), is calculated from, for each space i

$$hv = \sum_{j} \{ qin_{ij} * Ta_{j} + qout_{ij} * Ta_{i} \} * VSH ,$$

over each connecting space j

where qin_{ij} , $qout_{ij}$ are the airflows incoming from (+ve) and outgoing (-ve) to the space j , (m3/s),
Ta is the space air temperature (oC)
VSH is the volumetric specific heat of air.

The two flows, in and out, from one cell to another are treated separately as they will be equal only in the simplest cases. Net water transfer is treated similarily with the air water content (g/m3) taking the place o the air temperature in the above equation.

Note that the eternal node for air temperature and flows is represented by space 0, and that the 'virtual' spaces ,represented by space <0, may act as infinite sources and sinks of air if they are included in the specification of the airflow pattern.

VENTL - calculate ventilation transfer

Two airchange rates are calculated from the airflow data at each timestep, the total air change rate and the fresh air change rate. The total airchange rate is determined from the total air movement in the space i , as

$$\mathrm{ACRT} = \{ \sum_{j} (|qin|_{ij} + |qout|_{ij}) \}/2 \ / \ Vol_{i} * 3600 \quad ,$$

where Vol is the space volume (m3), and
ACRT is expressed in room volumes per hour.

The fresh air change rate is calculated from the air entry from the exterior only i.e.,

$$\mathrm{ACRF} = qin_{ij} / \ Vol \ {*}3600, \quad \text{for } j = 0 \ .$$

I/O:

Identification to default device if required.
Error and warning messages to default device.

ERROR TRAPPING:

Halts programme if unknown ventialtion option requested.

10.2 VENT1 - chose space air change rate - level 2

CALLED FROM : VENTL

CALL FORMAT : CALL VENT1

CALLS TO : -

CALL FREQUENCY : every timestep

GENERAL DESCRIPTION :

VENT1 determines the ventilating airflows in each space in the manner of the original ventilation model of HTB. Air exchange to the exterior only is considered, ie only fresh air change rates are specified, no internal flows. The airflow for each space is chosen from three specified rates according to a status switch and an air thermostat, or by override flags set via the diary.

THEORETICAL BASIS :

The final outcome of the routine is to fill the space-space flow matrix VFLOW. Only air entry form the exterior (space 0) is considered here, ther are no space to space flows specified.

This routine, in effect, simulates a simple three stage mechanical ventilation system with manual overrides, independant in each space. For automatic control the system is run through the ventilation status flag , as altered via the diary. When the ventialtion status is 'off' the first specified airchange rate is applied, ie as infiltration. When the ventilation status is 'on' then a choice between the two specified mechanical rates is made on the basis of the air temperature of the space in question. This may be summarised as,

1 status off - choose first rate
2 status on -
    a air temperature < set point - choose second rate
    b " " > " " - choose third rate.

This approach might, for instance, simulate a system where a high fan speed (or an open window) is used when the air temperature goes above a specified limit.

In addition a range of 'manual' overrides, again accessed via the diary, are available for each space as follows;

1 - first rate when system 'on'
2 - second rate " " "

3 - third rate " " " ,

the first rate is chosen when the system status is 'off'.

The following override the system status as well,

4 - first rate unconditionally
5 - second rate "
6 - third rate " .

I/O :

Identification to default device if required.
Error and warning messages to defualt device.

ERROR TRAPPING:

Error halts programme on ventilation selector out of known range (0-6).

10.3 VENT2 - chose space to space flow pattern - level 2

CALLED FROM : VENT1

CALL FORMAT : CALL VENT2

CALLS TO : -

CALL FREQUENCY : on demand , through flag VNTCHG

GENERAL DESCRIPTION :

VENT2 determines the ventilating airflows for each space to any other space chosen from a range of specified flow patterns. Internal cell to cell flows may be specified, and the virtual spaces may be included in the air flow patterns. Three different patterns may be specified and are chosen for use via the diary.

THEORETICAL BASIS :

The final outcome of the routine is to fill the space-space flow matrix VFLOW.

This routine offers the choice of three detailed interspace flow patterns, each available at any time according to the value of the decision flag. This approach may be of use when sufficiently rigid flow patterns exist as in a well sealed mechanical system, or where the inter space transport of heat and moisture is of interest.

Alterations to the flow matrix VFLOW are made only on request, using the flag VNTCHG. If an alteration is to be made then the appropriate flow pattern as chosen by the flag VNTOVR (using the first location) is copied into the final flow matrix. The flag VNTOVR is accessed via the diary.

I/O:

Identification to default device if required.
Error and warning messages to default device.

ERROR TRAPPING:

Error halts progrmme on ventilation selector out of range.

10.4 VENT3 - calculate empirical air change rate - level 2

CALLED FROM : VENTL

CALL FORMAT : CALL VENT3

CALLS TO : -

CALL FREQUENCY : on demand , though flags VNTCHG or METCHG

GENERAL DESCRIPTION:

VENT3 determines the ventilating airflows in each space, from a set of emperical airchange rate equations. Air exchange to the exterior only is considered, ie only fresh air change rates are specified, no internal flows. The equations are of a general form, based on wind speed and temperature difference between space and external air. A set of coefficients for these equations may be specified for each wind direction sector of 10 degrees.

THEORETICAL BASIS

The final outcome of the routine is to fill the space-space flow matrix VFLOW. Only air entry form the exterior (space 0) is considered here, ther are no space to space flows specified.

Past experience has shown that a usefully general equation for airchange rate can be described as

$$Acr = (A + B\sqrt{\Delta T} + C\ u) * D$$

where A,B,C,D are specified emperical coefficients ,

$\sqrt{\Delta T}$ is the square root of the space-exterior temperature difference,

u is the site wind speed (m/s).

This is the form of the equation used in this routine.

The coefficients A,B,C are specified for each 10 degree sector of wind direction and for each space. The coefficient D may act as an overall multiplier to account, for instance, for window opening, and may be accesed via the diary.

As in VENT2 the calculations of the ventilation rates are done only on demand, for instance when new meterological conditions are read.

This particular form for emperical equations for ventilation is not necessarily the most appropriate for all cases, however it is felt that its implementation here will act as a guide for others.

I/O:

Identification to default device if required.

10.5 VENT8 - use multicell mathematical model - level 2

CALLED FROM : VENTL

CALL FORMAT : CALL VENT8

CALLS TO : -

CALL FREQUENCY :

GENERAL DESCRIPTION:

VENT8 is at present a dummy routine. A multicell ventilation model is under testing for HTB2. This routine will abort the programme if called.

I/O:
Identification to default device if required.
Error and warning messages to default device.

ERROR TRAPPING:
Halts programme if dummy routine called.

CHAPTER 11

HEATING AND COOLING SYSTEMS

These routines deal with the calculation of the contribution of heating, (and cooling), systems to the energy balance of the building, and the response of associated controls.

HTSYS - determine system output 11-2
TSTAT - determine thermostat output 11-7
FROSTT - frost protection t'stat 11-13
SYSDEL - impose time delay on system output . . . 11-16
STTDEL - impose time delay on t'stat output . . . 11-19
CLSYS - determine cooling loads 11-22

11.1 HTSYS - determine system output - level 1

CALLED FROM : HTBOS

CALL FORMAT : CALL HTSYS

CALLS TO : TSTAT, FROSTT , SYSDEL

CALL FREQUENCY : every time-step

GENERAL DESCRIPTION

HTSYS determines the heating systems output in response to the current conditions of the building and of the system itself. The form of the system response characteristic is a exponential curve with specified time delay to control changes. Different time constant may be specified for warm-up and cool-down characteristics. Each system has an individual time-clock controller and "manual" overrides, and a thermostat control with specifiable characteristics. Frost protection thermostats are also available when system status is off. The net output from each system is divided into convective, radiative and direct (i.e. internal to a construction) outputs, and each of these may be apportioned to different spaces, surfaces, or elements by specified weighting characteristics. Refer to fig 11.1.

THEORETICAL BASIS

Each system has associated with it a 3 interval time clock control, a "manual" (i.e. set via diary) on/off override, a thermostat (and a frost stat), and a specification of convective, radiative and direc proportions and associated lists of connections and weightings to spaces, surfaces and elements. Each system is assumed to operate independantly.

The overall system status is determined from the time clocks and the overides.

Depending on this status a call is made to the thermostat routine or to the frost protection routine to determine the heat demand to the system. This heat demand is expressed as a fraction of the maximum heat available from the system, and depending on the control characteristic of the thermostat chosen (discussed in the thermostat module) may be a simple on/off signal, or a signal proportional to a temperature band or to the shortfall of heat content sensed.

The ability of the system to meet the demand is governed by the exponential response of the system. The response of the system is defined as

$$Ht = Hreq * (1 - e^{-t/\tau_w})$$

for the system warming up (a demand greater than that previously required) or

$$Ht = Hreq * (e^{-t/\tau_c})$$

for the system cooling down (a demand less than that previously required,

where Ht is the output at time t (W),
$Hreq$ is the required output (W),
τ_w, τ_c are the warm and cool time constants.

The output of the system is calculated using the previous output and the required output by

$$Ht = H' * Ef + Hreq * (1-Ef)$$

where H' is the previous time-step output,
Ef is a constant factor determined from

$$Ef = e^{- dt/\tau}$$

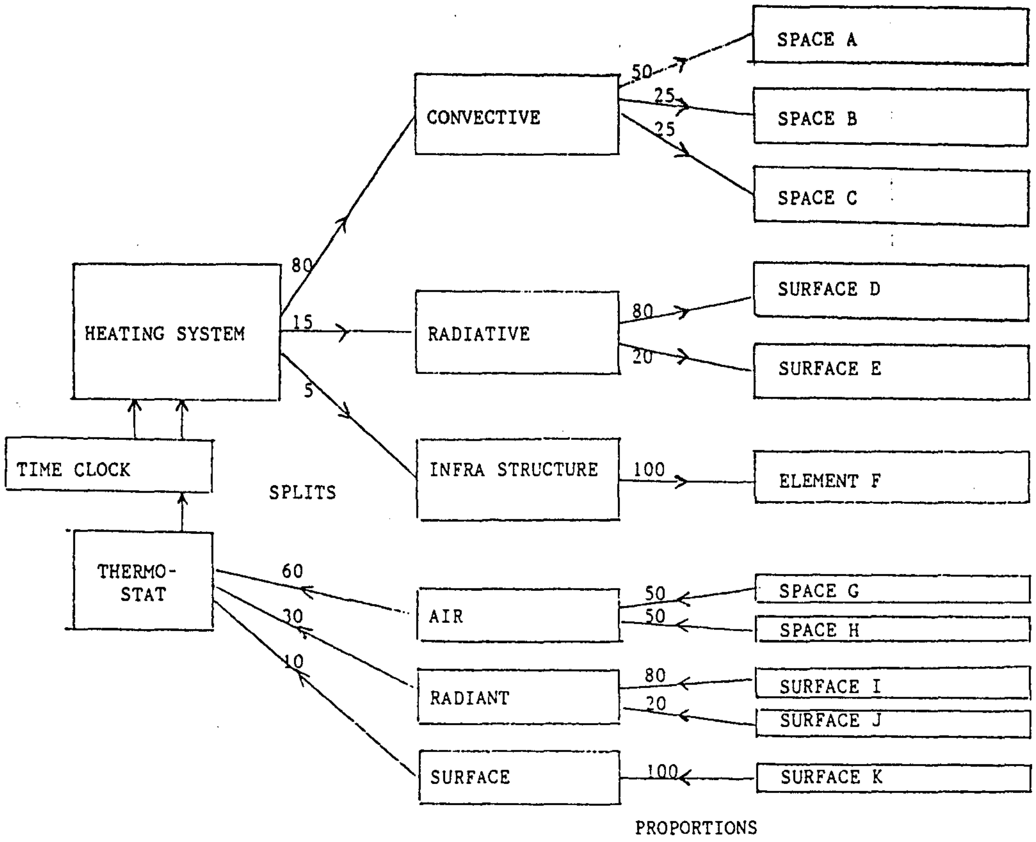
where dt is the calculation timestep, a separate factor is determined for warm-up and cool down.

The ability of the system to meet the required output can also be affected by a time lag to control changes. This is achieved by determining the response of the system to the current control requirement, as above, but storing this for future use on a stack structure and recalling the output stored one time lag period past for presentation as the current system output.

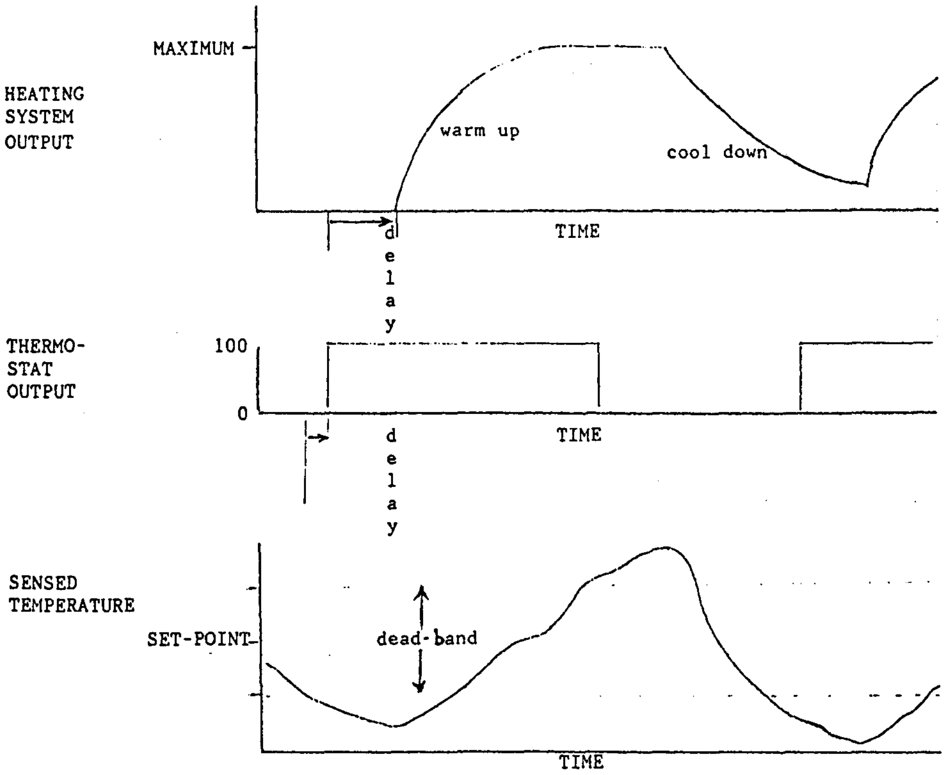
The current system output is split into the system specified convective, radiant, and direct components. The convective component thus determined is apportioned to a list of spaces to which the system has been specified as connected, by the associated weighting factors. This convective gain to each space is accumulated for each system for later application by the routine TEMPSP. Radiant gains to surfaces from the systems are determined in a similar manner to be applied in the routine TEMPRD (or EXCTRD). Direct gains are also accumulated for each calculatoin slice fo an element, for use as a direct heat gain within a structure by the finite differnce transfer routine FABRIC.

I/O:

Identification to default device if required.



A. CONNECTIVITIES



B. CHARACTERISTICS

fig 11.1 HEATING SYSTEM CHARACTERISTICS

11.2 TSTAT - determine thermostat output - level 2

CALLED FROM : HTSYS

CALL FORMAT : CALL TSTAT(I)
I > pointer to heat system , integer

CALLS TO : STTDEL

GENERAL DESCRIPTION:

TSTAT calculates thermostat control output for the heating system I, producing a signal for heat required from the system as a fraction of the maximum available from that system. The thermostat may be specified as having connections to various spaces and surfaces for to produce a response to a weighted mean temperature and may have defined coupling coefficients to air, radiant and surface temperatures. The thermostat may also be defined with an exponential response, an accelerator, and a time lag on output. The thermostat may produce simple on/off signal, or a signal proportional to a temperature band, or to the amount of heat needed to maintain a space temperature, depending on the option chosen at input.

THEORETICAL BASIS:

Each heating system has an independant thermostat sensor and control criterion associated with it.

The response of the thermostat is related to the temperature seen by the sensor (a mix of space, radiant, and surface temperatures), the reaction of the sensor to that temperature (an exponential time constant to changes), the effect on sensed temperature by an accelerator circuit (a superposed temperature with a separate time constant applied when system is on), a decision criteria (turn system on or off or vary output, and a time lag imposed on any decisions. Refer to figure 11.1.

The sensable temperature (ie. the temperature presented to the sensor) is calculated from weighted mean air, radiant, and surface temperatures as

$$Tsens = Ca * Tair + Cr * Trad + Cs * Tsurf$$

where Ca,Cr,Cs are coupling coefficients specified for each sensor. The weighted mean temperatures above are determined from specified lists of connections for each sensor.

Tair is a weighted mean of specified space air temperatures

$$Tair = \sum_i Tair_i * AW_i \quad \text{,for each specified space i.}$$

Trad is similarily a mean of either effective radiant temperatures at specified surfaces

$$Trad = \sum_i Tefrad_i * RW_i \quad \text{, for each specified surface i,}$$

or a mean of effective space radiant temperatures

$$Trad = \sum_i Tmrt_i * RW_i \quad \text{, for each specified zone i,}$$

depending on a option flag specified for each sensor.
Tsurf is similarily determined from surface temperatures

$$Tsurf = \sum_i Tsurface_i * SW_i \quad \text{, for each specified surface i.}$$

Aw , Rw , Sw are specifed weighting coefficients.

Normally a sensor would be connected to only one space and one surface so that the sensible temperature would be a combination of air temperature of that space, radiant temperature seen by the mounting surface, and surface temperature of the mounting surface. It may be desirable however to have a control on, say, the overall mean air temperature of selected spaces, which this procedure would allow.

The sensed temperature (i.e. that actually sensed) of the sensor is determined by an exponential response to temperature changes and calculated using the previous sensed temperature and current sensible temperature by

$$Tsensed_t = Tsensed_{t-1} * Ef + Tsens * (1 - Ef) \quad \text{at time t}$$

where Ef is a constant determined from

$$Ef = e^{(- tau/dt)} ,$$

tau is the time constant of the sensor (the same time constant is used for warmup and cooldown) and dt is the time interval used between calculations (timestep). This value is precalculable. A response characteristic of this type may be separately specified for each sensor.

The action of an optional accelerator is simulated by a superimposed temperature gain on the sensed temperature. This gain is determined from an exponential response of the accelerator to the system operation, when the system is "on" the accelerator is warming up to its maximum temperature addition, and when the system is off the accelorator cools towards zero temperature addition. Hence

$$Tsensed = Tsensed + Taccel,$$

where the Tsensed on the RHS has been calculated as above and

$$Taccel = Taccel_{t-1} * expaccel + Ttarget * (1- expaccel).$$

Expaccel is a constant determined form the accelerator time constant, similar to expcoef above, and Ttarget is either the maximum temperature increase of the accelerator, or zero, depending on the accelerator warming or cooling.

The response of the thermostat to the sensed temperature is calculated in two stages. First a decision on whether or not heat is required of the system, then a calculation of the amount of heat required.

The decision is made with regard to a specified set point temperature and a specified deadband range. If the sensed temperature is greater than the setpoint then no heat is required, if it is lower than the setpoint less the deadband then heat is required, and if it is within the deadband then the decision of the previous timeperiod is unchanged.

The calculation of the amount of heat required of the system depends on the mode of operation chosen for the particular system. In any case the output required is expressed as a fraction of the maximum output available from the system at the time of the call (note this is the maximum theoretical output , not the actual output which may be subjected to a time constant) and is in the range o 0 - 1. For a simpleon/off system the output required is either 1.0 or 0.0 (i.e. maximum or zero) depending on the decision taken as above. Alternatively a proportional output based on a temperature band may be calculated as

$$Output = (Tsetpoint - Tsensed)/ Bandwidth,$$

where Tsetpoint and Bandwidth are specified for each sensor using this mode. A third mode is available which determines the heat output required to maintain a set temperature in a space by calculating the shortfall of air heat content. The output is calculated as

$$Output = [(Tsetpoint - Tsensed) * vshc / timestep]/ maxheat,$$

where vshc is the volumetric specific heat capacity of the space and maxheat is the maximum heat available from the system. This mode is usable where the thermostat and system are connected to one space only.

Finally the thermostat/control output thus calculated may be subjected to a time delay before it is presented to the system. Similar to the method used in HTSYS for inducing time delays, the output is stored on a stack for future use, and a previously stored value retrieved for presentation to the current system calculations.

I/O :

Identification to default device if required.

11.3 FROSTT - frost protection t'stat - level 2

CALLED FROM : HTSYS

CALL FORMAT : CALL FROSTT(I)
I > heater system pointer , integer

CALLS TO : -

GENERAL DESCRIPTION

FROSTT is a simple frost protection thermostat, for the heating system IHT, which is invoked only when the system status is "off", ie. when normal control criteria do not apply. The thermostat is air temperature sensitive only, but may be a weighted average and can include the external air temperature. The output of the control, the fractional heat output required of the system, is either zero or a specified output for the protection system.

THEORETICAL BASIS

The sensed temperature is determined froma a weighted mean of specified space air temperatures (external air temperature may be accesed via space 0) as

$$\mathrm{Tsens} = \sum_i \mathrm{Tair}_i * \mathrm{Wa}_i \qquad \text{for each space i,}$$

where Wa is the weigthing for space i.

The control decision is made simply through comparison to a desired set point temperature. If the sensed temperature is greater then no heat is required form the system, if the sensed temperature is less than the set point then the maximum call allowed under the frost controller (a fraction of the maximum available from the system) is requested. There is no time constants or time delay on this controller.

I/O:

Identification to default device if required.

11.4 SYSDEL - impose time delay on system output - level 2

CALLED FROM : HTSYS

CALL FORMAT : CALL SYSDEL(I,L,V)

```
I > heating system pointer , integer
L > ring position pointer , integer
V > value to store on ring ,
 < previously stored value , real
```

CALLS TO : -

GENERAL DESCRIPTION

SYSDEL manages a ring stack to record and recall system output values to impose a time delay between the calculation and presentation of the system output. A separate ring is run for each system.

THEORETICAL BASIS

The entry value of V is stored on the stack at the location determined by the sack pointer RNGPNT. RNGPNT is incremented (modulus the stack size) each timestep by routine CLOCK so that stack locations before the current pointer represent values stored at past times. The entry value of L determines how many timesteps backward to retreive a past value. This value is now returned in the variable V. A separate ring stack is run for each heating system I.

I/O :

Identification to default device if required.

ERROR TRAPPING:

Warning on requested delay too large.

11.5 STTDEL - impose time delay on t'stat output - level 3

CALLED FROM : TSTAT

CALL FORMAT : CALL STTDEL(I,L,V)

```
I > heating system pointer , integer
L > ring position pointer , integer
V > value to store on ring ,
  < previously stored value , real
```

CALLS TO :

GENERAL DESCRIPTION

STTDEL manages a ring stack to record and recall thermostat output values to impose a time delay between the calculation and presentation of the thermostat output. A separate ring is run for each system.

THEORETICAL BASIS

The entry value of VALUE is stored on the stack at the location determined by the stack pointer RNGPNT. RNGPNT is incremented (modulus the stack size) each timestep by routine CLOCK so that stack locations before the current pointer represent values stored at past times. The entry value of LOC determines how many timesteps backward to retreive a past value. This value is now returned in the varaible VALUE. A separate ring stack is run for each heating system IHT.

I/O

Identification to default device if required.

ERROR TRAPPING:

Warning on requested delay too large.

11.6 CLSYS - calculate cooling load - level 1

CALLED FROM : HTBOS

CALL FORMAT : CALL CLSYS

CALLS TO : -

CALL FREQUENCY :

GENERAL DESCRIPTION:

CLSYS is a present a dummy routine for future inclusion of a cooling load sub-model. CLSYS will halt programme execution if called.

I/O:

Identification to default device if required.
Error and warning messages to default device.

ERROR TRAPPING:

Halts programme if called.

CHAPTER 12

INCIDENTAL GAINS

These routines deal with the management and calculation of incidental gains.

- CASUAL - manage incidental gain systems 12-2
  - OCCUPD - calculate occupants gains 12-5
  - SPOWER - calculate gains from power sources . . . 12-8
  - LIGHTS - calculate gains from light circuits . . . 12-11
    - LGHTMD - lighting load model 12-14

12.1 CASUAL - manage incidental gain systems - level 1

CALLED FROM : HTBOS

CALL FORMAT : CALL CASUAL

CALLS TO : OCCUPD , SPOWER , LIGHTS

CALL FREQUENCY : every timestep

GENERAL DESCRIPTION :

CASUAL is the managing module for the calculations of incidental gains. The incidental gains are divided into four independant source types ; occupants (biological sources), lighting, small power or miscellaneous sources, and an unassigned source type for users developement.

Each incidental gain process will independantly produce values for convective gain to spaces and raditive gain to surfaces. These values will be used in the calculation of new air temperature, and in the calculation of the net radiant exchange of surfaces. The level of the gains is primarily controlled through the diary structure.

There is an unused set of data arrays available for users development. These arrays represent convective gains to spaces (CNAUTG(space)), and radiant gains to surface (RDAUTG(element,side)). They are always included in the calculations but are maintained at zero unless changed by a users routine.

Each process may be disabled by a data flag.

I/O :

Identification to default device if required.

12.2 OCCUPD - calculate occupants gains - level 2

CALLED FROM : CASUAL

CALL FORMAT : CALL OCCUPD

CALLS TO : -

GENERAL DESCRIPTION :

OCCUPD accounts for the incidental gains from occupation (i.e. physiological heat output). The gain is related to occupancy level, and metabolic rate set for a particular space. The gain may be both radiant and convective. Water gains are also specifiable for occupants.

Each space has specified a range of represenational metabolic rates for heat and moisture output of an occupant of the space. These gain rates are multiplied by the current occupancy of the space to determine the net gain to the space. The number of occupants in a space, and the required metabolic rate, is accessed via the diary.

This total gain is apportioned firstly by dividing into radiative and convective components, from the specified occupancy split for that space. The convective portion of the heat gain, and the total water gain are accumlated for the space, while the radiative gain is further apportioned by surface view weighting factors specified.

I/O:

Identification to default device if required.

12.3 SPOWER - calculate gains from power sources - level 2

CALLED FROM : CASUAL

CALL FORMAT : CALL SPOWER

CALLS TO : -

GENERAL DESCRIPTION :

SPOWER manages the miscellaneous small power sources of heat and moisture. Each source may output to several spaces and surfaces according to its associated output level or an overide 'off' flag. Each source may have a separate radiative-covective output proportion. Output levels and the override flags are scheduled via the diary.

Each small power source is specified by its heat and water output, an override 'off' flag for switchable power, a radiative-convective output proportion, and a list of surfaces and view proportions for radiant output and of spaces ond output proportions for convective output.

The output at the time of calculation (which may be altered directly via the diary or indirectly by the override flag, also managed via the diary) is split into convective and radiative components. The convective component and the water output is apportioned to the list of connecting spaces by the weighting factors specified. The radiative component is apportioned to the list of connecting surfaces by the weighting factors.

I/O:

Identification to default device if required.

12.4 LIGHTS - calculate gains from light circuits - level 2

CALLED FROM : CASUAL

CALL FORMAT : CALL LIGHTS

CALLS TO : LGHTMD

GENERAL DESCRIPTION:

LIGHTS determines the incidental gains from lighting circuits. Each circuit is specified by a maximum output (equivalent to circuit load) and up to three fractional output levels governed by time-clock. A separate manual override fractional output, controlled from the diary, and an optional lighting load model are also available.

The output from each lighting circuit is determined according to the time of day, or a manual setting. There are three timed periods of lighting operation, each with a specified output. The manual operation also has a separate associated output level. An optional call to a lighting model may determine circuit output by other criteria, the supplied model switches between two maximum outputs according to external horizontal illumination(irradiance). The maximum output for each circuit is specified in kW, and each of the operational outputs is specified as a fraction of this maximum. Each of these values may of course be altered via the diary for more flexible specification of load profile.

The total output for each circuit is separated into radiative and convective components, which is then distributed to surfaces and spaces according to the specification of connections.

No water gains are assumed to originate from lighting.

I/O:

Identification to default device if required.

12.5 LGHTMD - lighting load model - level 3

```
CALLED FROM    : LIGHTS

CALL FORMAT    : CALL LGHTMD(I,O,S)
                         I > lighting circuit pointer , integer
                         O < fraction circuit load , real
                         S < circuit status , real

CALLS TO       : -

CALL FREQUENCY : conditional on procedure selected
```

GENERAL DESCRIPTION:

LGHTMD is a simple lighting model relating the maximum circuit load available to the external illuminance. It acts to modify the circuit loads determined in LIGHTS from time clocks and manual overides.

This simple lighting use model refers to the external horizontal irradiance, relating that to external illuminace, and determining which of two lighting regimes apply; i.e. night or day, according to a specified critical irradiance. If irradiance is greater than the specified limit for circuit I then the specified daytime circuit load factor is held to apply.

As net output is calculated to be a fraction of the total load specified for each circuit, the action of this model is to modify the scheduled fractional load as determined from time clocks or manual overrides. Thus for instance the time clock load may have been determined, and passed in O , as .80 but if the daytime load factor has been set to .50 (ie. only 1/2 maximum load available during daylight) then the net output would be returned in O as .40. The passed circuit status S is not yet used.

I/O:

Identification to default device if required.

CHAPTER 13

DIARY SCHEDULE INPUT

These routines deal with the operation of the scheduling diary..

DIARY - execute pending command 13-2
DRYNXT - queue next command 13-6
DRYIN - read next command from file 13-9

13.1 DIARY - execute pending command - level 1

CALLED FROM : HTBOS

CALL FORMAT : CALL DIARY

CALLS TO : DRYNXT , ZIPR6 , ZIPR7 , ZIPR8

GENERAL DESCRIPTION :

DIARY manages and executes commands contained in the current diary file. These commands may alter various data variables in the model database, or may alter option or operational variables of the model. The module is called by HTBOS when the current action is due, DIARY then parses the command, executes the command and determines the required time for the next command. The commands are in the form of a time (related to the time of day maintained by the model), a keyword for the action required, and the result of the action (typically a new value for a variable).

The database for the model is contained in various Fortran commons. Data in these are easily altered by this routine before the next round of calculations begin. Any desired action must have been previously identified and programmed into DIARY before it may be used. It is anticipated that the repretoire of commands will grow as HTB2 is used.

An external data file of diary commands is sequentially read and the individual commands executed when their specified time of action becomes due. The reading of the command line and the parsing of the command time are carried out in DRYNXT. This routine must be called initially to set up the first action, but DIARY uses DRTNXT to maintain the current action from then onwards.

Each diary file contains commands for one day. A list of which file, or 'page', is to be used on a particular day is specified in the diary list file. Commands in a diary file must be entered in time order. Several commands may be queued for the same time however, in which case they will be obeyed and implemented within one timestep.

Each command is a single line of ASCII data in a fairly rigid format, i.e.

```
HH:MM:SS   KEY   ACTION
```

where HH:MM:SS represents the time of day for this action to be executed,

KEY is the keyword for the action desired, and

ACTION is a character string which will be decoded by the particular action.

Usually the form of ACTION will be,

CODE VALUE,

where VALUE represents the new value for a variable, and
CODE indicates the type of alteration , possibly a pointer to an array location for the variable.

A list of commands currently available is found in the user manual.

The current command for execution resides in the location DRYLIN. DIARY first parses out the keyword for the action and transfers to the required procedure, if it has been defined (calls to undefined action procedures result in a warning message and a normal completion with no action taken).

It is the responsibilty of the procedure to parse the remaining information and alter the database as required. Also the procedure is responsible for maintaining the integrity of the database, ie. if a secondary variable has been calculated initially from the action variable it must be recalculated from the new value of the action variable.

The final operation in DIARY is to call DRYNXT to queue a new command in the comand buffer and determine a new time for the next call to DIARY from HTBOS.

I/O :

Identification to default device if required.
Error and warning messages to defualt device

ERROR TRAPPING:

Reports unknown commands and unreadable parameters.
Errors detected will halt programme if error recovery (LNOERR) is not enabled.

13.2 DRYNXT - queue next command - level 2

CALLED FROM : DIARY , INIT

CALL FORMAT : CALL DRYNXT

CALLS TO : DRYIN , CRASHR , FORERR

GENERAL DESCRIPTION :

DRYNXT retreives the next command line from the diary file, placing it in common buffer, and determines from that line the time for that command to occur, also returned in common.

DRYNXT uses the routine DRYIN to read a line of ascii text from a (previously opened) diary file. The command line is returned in the buffer DRYBUF and this is used as an internal file for the parsing operations of DRYNXT and DIARY. The command line is assumed to be in the fixed format hh:mm:ss cccccccccccccc..., where hh:mm:ss is a 24-hour time for the action to occur. DRYNXT extracts this time and calculates the equivalent time in seconds past midnight to be placed in the common variable RDTIME. This variable is used by HTBOS to determine when the next call to DIARY (to interpret the rest of the command in DRYBUF) is due. An error in extracting the time from the string is assumed to signal a comment and is ignored.

I/O :

Identification (if enabled) to default device.
Internal reading from the buffer DRYBUF.

ERROR TRAPPING:

Errors on reading the command line time are assumed to indicate a comment line. Line is ignored and a new line retrieved. No further action is taken.

Errors return from routine DRYIN are interpreted and acted upon. An error on read from DRYIN is taken as a fatal error, a message is output to default device and execution stopped. An end of file from DRYIN restes the diary flag so that no further calls to DIARY will be made (until externally reset, i.e. by openeing a new file at the start of a new day).

13.3 DRYIN - read next command from file - level 3

CALLED FROM : DRYNXT

CALL FORMAT : CALL DRYIN(L,C,S,E)
L < line buffer for command string , char
C < number characters read, integer
S > file stream for input , integer
E < error code, integer

CALLS TO : FORERR

GENERAL DESCRIPTION:

DRYIN retrieves a line of ascii text (132 characters maximum) from the file opened on unit S. The line is returned in L and the length of the string, not including trailing spaces, is returned in C. Error codes for success (0) , error on read(1), and end of file (2) are returned in E.

I/O:

Identification (if enabled) to default device.
Formatted, sequential reading of character data from external file.

ERROR TRAPPING:

Read error codes are returned to the calling routine. No action is taken in DRYIN.

CHAPTER 14

OUTPUT PROCEDURES

These routines deal with the output of data from HTB2. Data is output only if the appropriate routine has been enabled, either at input or via the diary.

REPORT - interval block output 14-2
LOGGER - data event logger 14-5
PRFOUT - element profile output 14-9

14.1 REPORT - interval block output - level 1

CALLED FROM : HTBOS

CALL FORMAT : CALL REPORT

CALLS TO : -

CALL FREQUENCY : at requested intervals

GENERAL DESCRIPTION:

If enabled at input, or at any time via the diary, REPORT produces a summary of selected topics on a selected interval. The data output represents accumulated averages over the preceding time interval. Intervals may be defined at input (defaulting to 60 minutes), and may be altered at any time, via the diary. Topics available are; Spaces , Water, Radiant, Element, Ventilation , Heating, and Meteorological, topics to be reported are selected at input. The data reported within each topic contains,

Spaces : air temp., total convective gains, convective gain breakdown from processes, for each space ,
Water : humidity , dewpoint , water gains from processes, for each space ,
Radiant: mean surface temp, mean rad temp, diffuse rad gains, for each zone ,
Element: surface temps and heat fluxes , direct radiant gains, for each element ,
Ventilation : fresh and total air change rates for each space,
Heating: output and duty cycle for each system,
Meteorological : average met data as supplied.

At present data is output as formatted, sequential text , to the file nominated at input.

I/O:

Identification to default device.
Formatted sequential output to unit BK1UNT.

14.2 LOGGER - data event logger - level 1

CALLED FROM : HTBOS

CALL FORMAT : LOGGER

CALLS TO : -

CALL FREQUENCY : every timestep when enabled

GENERAL DESCRIPTION :

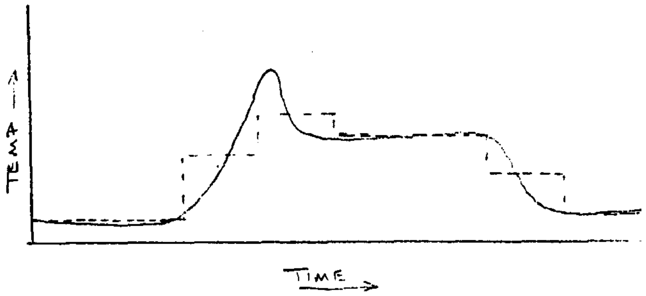
If enabled at input, or at any time via the diary, LOGGER monitors selected items and outputs data when their values has significantly changed. Thus sufficient data is produced to enabled the complete time history of an item to be reconstructed, without the need for large amounts of redundant information. Data may be recorded at each timestep, or only once a day, depending on the instantaneous rate of change of the item. Data recorded for an item is the item identifier, the time of the report, the item value of the previous timestep, and the current value. The items to monitor and the resolution (or level of a 'significant' change) are specified at input. Current items available include air temperature, readiant temperature, surface temperatures, heating systyem output, etc. A complete list is given in the user manual.

Currently data is output as formatted, sequential records.

As an example, consider the monitoring of air temperature in a space. The output records may be,

```
   :
19:05    16.5
19:07    17.0
19:10    17.5
19:13    18.0
19:15    18.5
19:17    19.0
19:18    18.5
19:20    18.0
19:25    17.5
   :
  etc.
```

which would allow the time history to be reconstructed as follows (with the equivalent hourly average report overlaid in a dotted line).



I/O:

Identification to default device.
Formatted sequential records to unit LOGUNT

14.3 PRFOUT - element profile output - level 1

CALLED FROM : HTBOS

CALL FORMAT : CALL PRFOUT

CALLS TO : -

CALL FREQUENCY : at requested intervals

GENERAL DESCRIPTION:

If enabled at input, or at any time via the diary, PRFOUT produces a report on the state of selected elements on a selected interval. The data output represents instantaneous values at the time of the report, and includes data on the surface environment (air and radiant temperatures, irradiation), and surface and interior temperatures and heat fluxes, for each node of the element.

Data is currently output in formatted sequential records.

I/O:
Identification to default device if required.
Formatted sequential records to unit BK2UNT.

CHAPTER 15

USER LINK POINTS - ZIPPERS

These routines provide an interface for users routines. There are five points placed at strategic areas within HTBOS;

1. after data input and initialisation, but before main calculation loop,
2. after initial condition calculation,
3. after meteorological and solar calculations,
4. after incidental gain calculations, and
5. after output procedures,

three points which can be called from the diary as one-shot routines, and a link point in the ventilation routine VENTL (with an associated input routine).

These routines may be connected to an otherwise unused common block, which is accessable through the diary, and to two filenames which are defined in the main input stage.

It is envisaged that these routines would be used to call the user routines, rather than being developed in themselves.

ZIPR1 15-2
ZIPR2 15-2
ZIPR3 15-2
ZIPR4 15-2
ZIPR5 15-2
ZIPR6 15-9
ZIPR7 15-9
ZIPR8 15-9
ZPRVNT 15-14
ZPRVNR 15-14

15.1 ZIPR1,2,3,4,5 - link points - level 1

CALLED FROM : HTBOS

CALL FORMAT : CALL ZIPR1 , etc.

CALLS TO : { for development for call to user routine }

CALL FREQUENCY : once enabled by option flags;

ZIPR1 - once at initialisation,

ZIPR2,3,4,5 - every time-step

GENERAL DESCRIPTION

These routines provide linkage points for inserting user subroutines at predefined locations within the calculations of HTB2. They are each selectively called according to their associated flags LZPR1,2,3,4,5. These flags are set according to the option choices made in the data input, or by commands through the diary. By default, they are not called. Once enabled, they will be executed every timestep until disabled.

As supplied, these are simply dummy routines, that is the return directly on being called. Should the insertion of a user routine be desired, the appropriate link routine should simply call the user routine. It is the responsibility of the user routine to maintain the validity of the database.

15.2 ZIPR6 , 7 , 8 - level 2

CALLED FROM : DIARY

CALL FORMAT : CALL ZIPR6(PARBUF) , etc.
: PARBUF - char - character string passed from diary command

CALLS TO : -

CALL FREQUENCY : on demand from Diary

GENERAL DESCRIPTION:

These zipper routines are located within the Diary structure. They can be called at any time during a run through diary commands. The are executed once only, in the timestep they were enabled, per call.

These routines are called with a character parameter string, which is specified in the diary command, and which could be passed to the user routine for processing.

15.3 ZPRVNT , ZPRVNR - ventilation link points - level 2

CALLED FROM : VENTL , (RDVENT)

CALL FORMAT : CALL ZPRVNT , CALL ZPRVNR(IERR)
: IERR - intg - return error code : -1 - error on read
: 0 - success

CALLS TO : -

CALL FREQUENCY : every timestep

GENERAL DESCRIPTION:

These zipper routines provide link points for alternate ventilation submodels, ZPRVNT can replace VENT1 , VENT2 , VENT3 , VENT8, while ZPRVNR provides a link point at input stage for specific data which may be required by ZPRVNT.

CHAPTER 16

GENERAL UTILITY ROUTINES

These routines deal with various general purpose matters.

CRASHR - abort run due to errors 16-2
RH2DP - convert RH to dewpoint 16-5
RH2VP - convert RH to vapour pressure . . . 16-8
RH2WC - convert RH to water content . . . 16-11
T2SVP - air temp. to saturated vap. press 16-14
VALDLN - take data line from file 16-17
VP2WC - vapour pressure to water content 16-20
WC2RH - water content to r.h. 16-23
WHTSIZ - report data totals 16-26

16.1 CRASHR - abort run due to errors - level U

CALL FORMAT : CALL CRASHR

CALLS TO : -

GENERAL DESCRIPTION:

CRASHR aborts a simulation run tidily, closing all files and reporting the internal time run was aborted.

I/O:

Identification to default device if required.
Information message to default device.

16.2 RH2DP - convert r.h. to dewpoint - level U

CALL FORMAT : RH2DP(Rh,T) (function call)
: > Rh relative humidity of air (%) , real
> T temperature of air (oC) , real

CALLS TO : RH2VP

GENERAL DESCRIPTION:

RH2DP converts relative humidity of air (Rh) of air at temperature (T), to a dewpoint temperature (oC).

THEORETICAL BASIS:

Dewpoint is considered to occur at the temperature when present water content, as defined by Rh, T, reaches saturation.

$$Dp = \frac{170.5868 - 258.15 * \mathrm{LOG}(Vp)}{1.15 * \mathrm{LOG}(Vp) - 9.05392}$$

where Vp is the vapour pressure (mmHg) as determined in routine RH2VP.

I/O:

Identification to default device if required.

16.3 RH2VP - convert RH to vapour pressure - level U

```
CALL FORMAT     : RH2VP(Rh,T) (function call)
                   : > Rh  relative humidity of air (%) , real
                     > T   temperature of air (oC) , real

CALLS TO        : T2SVP
```

GENERAL DESCRIPTION:

RH2VP converts relative humidity (Rh) at air temperature (T) to vapour pressure (mmHg).

THEORETICAL BASIS:

Vapour pressure is determined from,

$$Vp = Rh/100 * Svp ,$$

where Svp is the saturated vapour pressure for air at temperature T, as determined in routine T2SVP.

I/O:

Identification to default device if required.

16.4 RH2WC - convert RH to water content - level U

```
CALL FORMAT       : RH2WC(Rh,T)  (function call)
                     :  > Rh   relative humidity of air (%) , real
                        > T    temperature of air (oC) , real

CALLS TO          : T2SVP , VP2WC
```

GENERAL DESCRIPTION:

RH2WC converts relative humidity (Rh) at air temperature (T) to water content of air (g/m3).

THEORETICAL BASIS:

Water content is determined from,

Wc = Rh/100 * Swc

where Swc is the saturated water content (g/m3) at air temperature T as calculated by routines T2SVP , VP2WC.

I/O:

Identification to default device if required.

16.5 T2SVP - temperature to saturated vap. press - level 3

CALL FORMAT : T2SVP(T) (function call)
T > air temperature (oC) , real
returns saturated vapour pressure (mmHg), real

CALLS TO : -

GENERAL DESCRIPTION

T2SVP determines the saturated vapour pressure (mmHg) at a given air temperature (oC).

THEORETICAL BASIS

The saturated vapour pressure is calculated from,

$$\mathrm{Svp} = 10^{\left\{ 2.8808 - \dfrac{5.741 * (100 - T)}{258.15 + 1.15 * T} \right\}}$$

,where Svp is the saturated vapour pressure (mmHg) and
T is the air temperature (oC).

I/O:
Identification to default device if required.

16.6 VALDLN - take data line from file - level U

```
CALL FORMAT      : CALL VALDLN(UNIT,LINE,COUNT,LENGTH,ERR)
                    :  > UNIT   - unit number for input , integer
                       < LINE   - character record returned , char
                       < COUNT  - number of record read , integer
                       < LENGTH - number of characters in record, intg
                       < ERR    - error code  = 0 - success
                                              -1 - end of file
                                             <-1 - read error code

CALLS TO         : -
```

GENERAL DESCRIPTION:

VALDLN accepts a record of character data from a file connected to the passed unit. VALDLN determines if the record is a valid data record (i.e. not of null length, or not starting with a comment character '*'), and if so returns it and its length. If the reocrd is not valid, new records are read until a valid line is found, or until an error condition is reached.

VALDLN strips leading spaces from the record , and determines the length to the last non-space character, or to a comment character.

I/O:

Identification to default device if required.
Formatted sequential input from passed unit.

ERROR TRAPPING:

Detects and returns fortran read errors, including end of file reached. No further action is taken in VALDLN.

16.7 VP2WC - vapour pressure to water content - level 3

CALL FORMAT : VP2WC(V) function call
V > vapour pressure (mmHg), real
returns water content (g/m3), real

CALLS TO : -

GENERAL DESCRIPTION

WC2RH determines the equivalent water content for the vapour pressure presented.

THEORETICAL BASIS

The relationship between vapour pressure and water content used is,

$$Wc = 0.94468 * V + 0.55473 ,$$

where V is the vapour pressure (mmHg) and
Wc is the water content (g/m3).

I/O:
Identification to default device if required.

16.8 WC2RH - water content to r.h. - level 2

```
CALL FORMAT    : WC2RH(T,W)  -function call
                 T > air temp (oC), real
                 W > water content (g/m3), real
                 returns relative humidity (%), real
CALLS TO       : VP2WC , T2SVP
```

GENERAL DESCRIPTION

WC2RH produces the equivalent relative humidity for the water content and air temperature given.

THEORETICAL BASIS

The relative humidity is determined as

Rh = 100.0 * W / Swc

, where W is the air water content (g/m3) and
SwC is the saturated water content at the air temperature given (g/m3), as detemined from T2SVP , VP2WC.

I/O:

Identification to default device if requested.

16.9 WHTSIZ - report data totals - level U

CALL FORMAT : CALL WHTSIZ

CALLS TO : -

GENERAL DESCRIPTION:

WHTSIZ reports on the data-base size and usage at the input stage of HTB2. On the first call WHTSIZ reports the sizing parameters used in building the programme in use, while on second call the data totals read at input are reported.

I/O:

Identification to default device if required.
Information messages to default device.

CHAPTER 17

MACHINE/COMPILER SPECIFIC ROUTINES

These routines isolate procedures which may require syntax, or system routines, specific to a particular computer or compiler.

In this documentation both VAX/VMS and PRIME/PRIMOS/FTN77 versions will be given.

DSTAMP - get system date and time 17-2
FORERR - report system error codes 17-6
OPBLK1 - open report file 17-10
OPBLK2 - open profile file 17-14
OPLGR1 - open event logger file 17-18
OPNBLD - open buildings file 17-22
OPNCON - open construction file 17-26
OPNDRY - open diary page file 17-30
OPNDYL - open diary list file 17-34
OPNHTR - open heating system file 17-38
OPNINF - open information file 17-42
OPNLAY - open fabric layout file 17-46
OPNLGT - open lighting system file 17-50
OPNMAT - open materials library file . . . 17-54
OPNMET - open meteorologocal file 17-58
OPNOCC - open occupancy file 17-62
OPNSPW - open small power file 17-66
OPNSRV - open services file 17-70
OPNTOP - open top level file 17-74
OPNVNT - open ventilation file 17-78
OPUMAT - open users materials file 17-82

17.1 DSTAMP - get system date and time - level M

CALLED FROM	: INIT
CALL FORMAT	: CALL DSTAMP
CALLS TO	: (system routines)
CALL FREQUENCY	: once

GENERAL DESCRIPTION:

DSTAMP is a machine or installation specific routine to retrieve the system date and time using routines that may be available though system utility libraries. DSTAMP also provides an identification string containing model version and machine name.

DSTAMP is called at startup and is not critical to the operation of HTB2 if such system routines are not available.

I/O:

Identification to default device if required.

17.2 FORERR - report system error codes - level M

```
CALL FORMAT      : CALL FORERR(IERR)
                     : > IERR - intg - system error code
CALLS TO         : (system routines)
```

GENERAL DESCRIPTION:

FORERR is a general purpose routine to translate system fortran error codes (primarily I/O errors) into more understandable messages.

I/O:

Identification to default device if required.

17.3 OPBLK1 - open report file - level M

```
CALLED FROM       : INPUT

CALL FORMAT       : CALL OPBLK1(FILE,ERR)
                            > FILE  - file name - char
                            < ERR   - open error code - intg
                                   =0 success
                                    -1 failure

CALLS TO          : -

CALL FREQUENCY    : once
```

GENERAL DESCRIPTION:

OPBLK1 opens the nominated file for output by REPORT, on unit BK1UNT. This file is opened at input and remains active until the end of run. Currently the file type is formatted, sequential.

The PRIME version of this routine appends a version code to the passed file name, and will thus attempt to open a new file if the required name is already in use. Up to 26 versions are attempted before failure. VAX/VMS automatically append version numbers to filenames.

If a file is successfully opened the full file specification filename is returned in the calling arguement.

I/O:

Identification to default device if required.
Error messages to default device.
File operations on unit BK1UNT.

ERROR TRAPPING:

Errors in opening nominated file are reported , no further action is taken.

17.4 OPBLK2 - open profile file - level M

```
CALLED FROM       : INPUT

CALL FORMAT       : CALL OPBLK2(FILE,ERR)
                            > FILE  - file name - char
                            < ERR   - open error code - intg
                                   =0 success
                                    -1 failure
CALLS TO          : -

CALL FREQUENCY    : once
```

GENERAL DESCRIPTION:

OPBLK2 opens the nominated file for output by PRFOUT, on unit BK2UNT. This file is opened at input and remains active until the end of run. Currently the file type is formatted, sequential.

The PRIME version of this routine appends a version code to the passed file name, and will thus attempt to open a new file if the required name is already in use. Up to 26 versions are attempted before failure. VAX/VMS automatically append version numbers to filenames.

If a file is successfully opened the full file specification filename is returned in the calling arguement.

I/O:

Identification to default device if required.
Error messages to default device.
File operations on unit BK2UNT.

ERROR TRAPPING:

Errors in opening nominated file are reported , no further action is taken.

17.5 OPLGR1 - open event log file - level M

```
CALLED FROM      : INPUT

CALL FORMAT      : CALL OPLGR1(FILE,ERR)
                           > FILE  - file name - char
                           < ERR   - open error code - intg
                                 =0 success
                                  -1 failure
CALLS TO         : -

CALL FREQUENCY   : once
```

GENERAL DESCRIPTION:

OPLGR1 opens the nominated file for output by LOGGER, on unit LOGUNT. This file is opened at input and remains active until the end of run. Currently the file type is formatted, sequential.

The PRIME version of this routine appends a version code to the passed file name, and will thus attempt to open a new file if the required name is already in use. Up to 26 versions are attempted before failure. VAX/VMS automatically append version numbers to filenames.

If a file is successfully opened the full file specification filename is returned in the calling arguement.

I/O:

Identification to default device if required.
Error messages to default device.
File operations on unit LOGUNT.

ERROR TRAPPING:

Errors in opening nominated file are reported , no further action is taken.

17.6 OPNBLD - open building file - level M

```
CALLED FROM       : DEFBLD

CALL FORMAT       : CALL OPNBLD(FILE,ERR)
                            > FILE  - file name - char
                            < ERR   - open error code - intg
                                   =0 success
                                    -1 failure
CALLS TO          : -

CALL FREQUENCY    : once
```

GENERAL DESCRIPTION:

OPNBLD opens the nominated file for input for DEFBLD. Unit INPUNT is used, the file remains connected until all data is read. The file type is formatted sequential.

The full file specification name is returned if opened successfully.

Where installations allow the file should be opened with READONLY status, and allow multiple connections.

I/O:

Identification to default device if required.
Error messages to default device.
File operations on unit INPUNT.

ERROR TRAPPING:
File errors on opening are reported , no further action is taken
in this routine.

17.7 OPNCON - open construction file - level M

```
CALLED FROM       : RDCON

CALL FORMAT       : CALL OPNCON(FILE,ERR)
                          > FILE  - file name - char
                          < ERR   - open error code - intg
                                 =0 success
                                  -1 failure
CALLS TO          : -

CALL FREQUENCY    : once
```

GENERAL DESCRIPTION:

OPNCON opens the nominated file for input for RDCON. Unit INPUNT is used, the file remains connected until all data is read. The file type is formatted sequential.

The full file specification name is returned if opened successfully.

Where installations allow the file should be opened with READONLY status, and allow multiple connections.

I/O:

Identification to default device if required.
Error messages to default device.
File operations on unit INPUNT.

ERROR TRAPPING:
File errors on opening are reported , no further action is taken in this routine.

17.8 OPNDRY - open diary file - level M

```
CALLED FROM       : NEWDAY , INPUT

CALL FORMAT       : CALL OPNDRY(FILE,ERR)
                            > FILE  - file name - char
                            < ERR   - open error code - intg
                                   =0 success
                                    -1 failure
CALLS TO          : -

CALL FREQUENCY    : once daily
```

GENERAL DESCRIPTION:

OPNDRY opens the current dairy page file for input for DRYLIN. Unit DRYUNT is used, the file remains connected unit next page is required. The file type is formatted sequential.

The full file specification name is returned if opened successfully.

Where installations allow the file should be opened with READONLY status, and allow multiple connections.

I/O:

Identification to default device if required.
Error messages to default device.
File operations on unit DRYUNT.

ERROR TRAPPING:
File errors on opening are reported , no further action is taken
in this routine.

17.9 OPNDYL - open diary list file - level M

```
CALLED FROM       : INPUT

CALL FORMAT       : CALL OPNDYL(FILE,ERR)
                             > FILE  - file name - char
                             < ERR   - open error code - intg
                                    =0 success
                                     -1 failure
CALLS TO          : -

CALL FREQUENCY    : once
```

GENERAL DESCRIPTION:

OPNDYL opens the nominated diary page list file for input for NEWDAY. Unit DYLUNT is used, the file remains connected until th eend of run. The file type is formatted sequential.

The full file specification name is returned if opened successfully.

Where installations allow the file should be opened with READONLY status, and allow multiple connections.

I/O:

Identification to default device if required.
Error messages to default device.
File operations on unit DYLUNT.

ERROR TRAPPING:
File errors on opening are reported , no further action is taken
in this routine.

17.10 OPNHTR - open heating system file - level M

```
CALLED FROM       : RDHTR

CALL FORMAT       : CALL OPNHTR(FILE,ERR)
                            > FILE  - file name - char
                            < ERR   - open error code - intg
                                   =0 success
                                    -1 failure
CALLS TO          : -

CALL FREQUENCY    : once
```

GENERAL DESCRIPTION:

OPNHTR opens the nominated file for input for RDHTR. Unit INPUNT is used, the file remains connected until all data is read. The file type is formatted sequential.

The full file specification name is returned if opened successfully.

Where installations allow the file should be opened with READONLY status, and allow multiple connections.

I/O:

Identification to default device if required.
Error messages to default device.
File operations on unit INPUNT.

ERROR TRAPPING:
File errors on opening are reported , no further action is taken in this routine.

17.11 OPNINF - open information file - level M

```
CALLED FROM       : INFO

CALL FORMAT       : CALL OPNINF(FILE,ERR)
                            > FILE  - file name - char
                            < ERR   - open error code - intg
                                   =0 success
                                    -1 failure

CALLS TO          : -

CALL FREQUENCY    : once
```

GENERAL DESCRIPTION:

OPNINF opens the nominated file for output by INFO, on unit INFUNT. This file is open only within the execution of the routine at input stage. The file type is formatted, sequential.

The PRIME version of this routine appends a version code to the passed file name, and will thus attempt to open a new file if the required name is already in use. Up to 26 versions are attempted before failure. VAX/VMS automatically append version numbers to filenames.

If a file is successfully opened the full file specification filename is returned in the calling arguement.

I/O:

Identification to default device if required.
Error messages to default device.
File operations on unit INFUNT.

ERROR TRAPPING:

Errors in opening nominated file are reported , no further action
is taken.

17.12 OPNLAY - open layout file - level M

```
CALLED FROM       : RDLAY

CALL FORMAT       : CALL OPNLAY(FILE,ERR)
                            > FILE  - file name - char
                            < ERR   - open error code - intg
                                   =0 success
                                    -1 failure
CALLS TO          : -

CALL FREQUENCY    : once
```

GENERAL DESCRIPTION:

OPNLAY opens the nominated file for input for RDLAY. Unit INPUNT is used, the file remains connected until all data is read. The file type is formatted sequential.

The full file specification name is returned if opened successfully.

Where installations allow the file should be opened with READONLY status, and allow multiple connections.

I/O:

Identification to default device if required.
Error messages to default device.
File operations on unit INPUNT.

ERROR TRAPPING:
File errors on opening are reported , no further action is taken in this routine.

17.13 OPNLGT - open lighting system file - level M

CALLED FROM : RDLGT

CALL FORMAT : CALL OPNLGT(FILE,ERR)

```
> FILE  - file name - char
< ERR   - open error code - intg
       =0 success
        -1 failure
```

CALLS TO : -

CALL FREQUENCY : once

GENERAL DESCRIPTION:

OPNLGT opens the nominated file for input for RDLGT. Unit INPUNT is used, the file remains connected until all data is read. The file type is formatted sequential.

The full file specification name is returned if opened successfully.

Where installations allow the file should be opened with READONLY status, and allow multiple connections.

I/O:

Identification to default device if required.
Error messages to default device.
File operations on unit INPUNT.

ERROR TRAPPING:
File errors on opening are reported , no further action is taken in this routine.

17.14 OPNMAT - open materials library file - level M

```
CALLED FROM       : DEFBLD

CALL FORMAT       : CALL OPNMAT(UNIT,FILE,ERR)
                            > UNIT  - i/o unit - intg
                            > FILE  - file name - char
                            < ERR   - open error code - intg
                                   =0 success
                                    -1 failure
CALLS TO          : -
```

GENERAL DESCRIPTION:

OPNMAT opens the nominated materials library file for input for DEFBLD. The file remains connected only for one read. The file type is formatted, direct access.

The full file specification name is returned if opened successfully.

Where installations allow the file should be opened with READONLY status, and allow multiple connections.

I/O:

Identification to default device if required.
Error messages to default device.
File operations on unit nominated.

ERROR TRAPPING:
File errors on opening are reported , no further action is taken
in this routine.

17.15 OPNMET - open meteorological file - level M

```
CALLED FROM      : INPUT

CALL FORMAT      : CALL OPNMET(FILE,ERR)
                          > FILE  - file name - char
                          < ERR   - open error code - intg
                                =0 success
                                 -1 failure
CALLS TO         : -

CALL FREQUENCY   : once
```

GENERAL DESCRIPTION:

OPNMET opens the nominated file for input for METBAS. Unit METUNT is used, the file remains connected until end of run. The file type is formatted sequential.

The full file specification name is returned if opened successfully.

Where installations allow the file should be opened with READONLY status, and allow multiple connections.

I/O:

Identification to default device if required.
Error messages to default device.
File operations on unit METUNT.

ERROR TRAPPING:
File errors on opening are reported , no further action is taken in this routine.

17.16 OPNOCC - open occupancy file - level M

CALLED FROM : RDOCC

```
CALL FORMAT      : CALL OPNOCC(FILE,ERR)
                          > FILE  - file name - char
                          < ERR   - open error code - intg
                                 =0 success
                                  -1 failure
```

CALLS TO : -

CALL FREQUENCY : once

GENERAL DESCRIPTION:

OPNOCC opens the nominated file for input for RDOCC. Unit INPUNT is used, the file remains connected until all data is read. The file type is formatted sequential.

The full file specification name is returned if opened successfully.

Where installations allow the file should be opened with READONLY status, and allow multiple connections.

I/O:

Identification to default device if required.
Error messages to default device.
File operations on unit INPUNT.

ERROR TRAPPING:
File errors on opening are reported , no further action is taken in this routine.

17.17 OPNSPW - open small power file

```
CALLED FROM       : RDSPW

CALL FORMAT       : CALL OPNSPW(FILE,ERR)
                            > FILE  - file name - char
                            < ERR   - open error code - intg
                                   =0 success
                                    -1 failure
CALLS TO          : -

CALL FREQUENCY    : once
```

GENERAL DESCRIPTION:

OPNSPW opens the nominated file for input for RDSPW. Unit INPUNT is used, the file remains connected until all data is read. The file type is formatted sequential.

The full file specification name is returned if opened successfully.

Where installations allow the file should be opened with READONLY status, and allow multiple connections.

I/O:

Identification to default device if required.
Error messages to default device.
File operations on unit INPUNT.

ERROR TRAPPING:
File errors on opening are reported , no further action is taken in this routine.

17.18 OPNSRV - open services file - level M

```
CALLED FROM       : DEFSRV

CALL FORMAT       : CALL OPNSRV(FILE,ERR)
                            > FILE  - file name - char
                            < ERR   - open error code - intg
                                   =0 success
                                    -1 failure
CALLS TO          : -

CALL FREQUENCY    : once
```

GENERAL DESCRIPTION:

OPNSRV opens the nominated file for input for DEFSRV. Unit INPUNT is used, the file remains connected until all data is read. The file type is formatted sequential.

The full file specification name is returned if opened successfully.

Where installations allow the file should be opened with READONLY status, and allow multiple connections.

I/O:

Identification to default device if required.
Error messages to default device.
File operations on unit INPUNT.

ERROR TRAPPING:
File errors on opening are reported , no further action is taken
in this routine.

17.19 OPNTOP - open top-level file - level M

CALLED FROM : INPUT

CALL FORMAT : CALL OPNTOP(ERR)
< ERR - open error code - intg
=0 success
-1 failure

CALLS TO : -

CALL FREQUENCY : once

GENERAL DESCRIPTION:

OPNTOP opens the top-level file for input for INPUT. Unit TOPUNT is used, the file remains connected until input stage is finished. The file type is formatted sequential.

The file used for input is preconnected, in the case of a PRIME installation, on the system unit corresponding to TOPUNT. On a VAX/VMS installation the top-level file name is nominated though an assignment to the variable HTBCMD.

The full file specification name is returned if opened successfully.

Where installations allow the file should be opened with READONLY status, and allow multiple connections.

I/O:

Identification to default device if required.
Error messages to default device.
File operations on unit INPUNT.

ERROR TRAPPING:
File errors on opening are reported , no further action is taken in this routine.

17.20 OPNVNT - open ventilation file - level M

```
CALLED FROM       : RDVENT

CALL FORMAT       : CALL OPNVNT(FILE,ERR)
                            > FILE  - file name - char
                            < ERR   - open error code - intg
                                  =0 success
                                   -1 failure
CALLS TO          : -

CALL FREQUENCY    : once
```

GENERAL DESCRIPTION:

OPNVNT opens the nominated file for input for RDVENT. Unit INPUNT is used, the file remains connected until all data is read. The file type is formatted sequential.

The full file specification name is returned if opened successfully.

Where installations allow the file should be opened with READONLY status, and allow multiple connections.

I/O:

Identification to default device if required.
Error messages to default device.
File operations on unit INPUNT.

ERROR TRAPPING:
File errors on opening are reported , no further action is taken in this routine.

17.21 OPUMAT - open users materials file - level M

```
CALLED FROM       : DEFBLD

CALL FORMAT       : CALL OPUMAT(UNIT,FILE,ERR)
                            > UNIT  - i/o unit - intg
                            > FILE  - file name - char
                            < ERR   - open error code - intg
                                   =0 success
                                    -1 failure
CALLS TO          : -
```

GENERAL DESCRIPTION:

OPUMAT opens the nominated users materials file for input for DEFBLD. The file remains connected until all data is read. The file type is formatted sequential.

The full file specification name is returned if opened successfully.

Where installations allow the file should be opened with READONLY status, and allow multiple connections.

I/O:

Identification to default device if required.
Error messages to default device.
File operations on unit nominated.

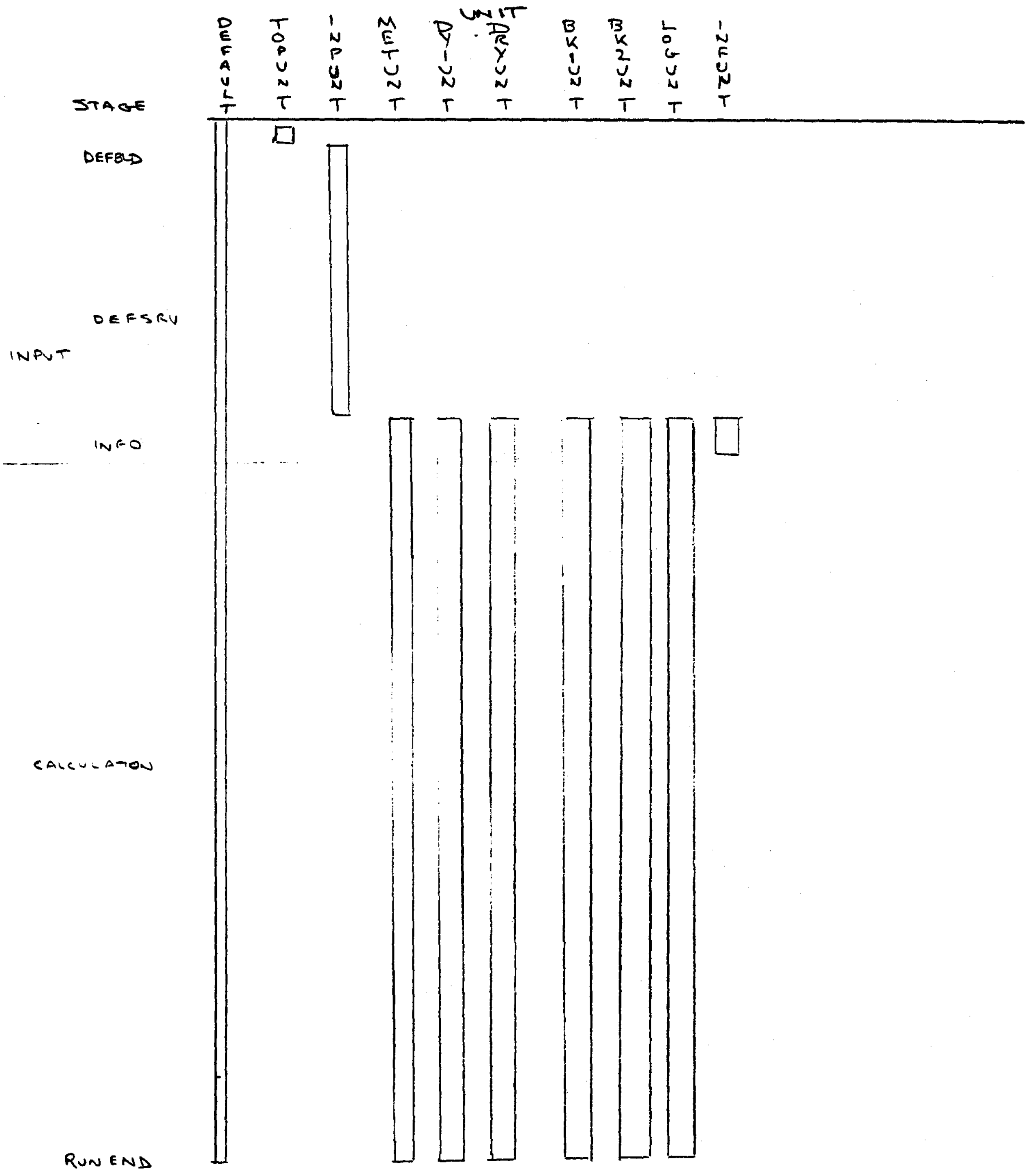
ERROR TRAPPING:
File errors on opening are reported , no further action is taken in this routine.

APPENDIX 2 - I/O UNIT USAGE

1.0 I/O Unit Usage

This appendix describes the usage of FORTRAN I/O units through the body of HTB2. All I/O is done through units defined in integer variables, such as BK1UNT or INPUNT. The actual unit number used is defined at compilation by the included file 'PARDEF.CMN', this file may be modified to overcome possible conflicts or limitations in some installatons. The only exception to this is the use of the FORTRAN default device ('*') for error and information messages to the running terminal or batch log file, this unit is usually predefined for a specific installation, note that is is important that the actual unit number used by a system for this I/O must not be in conflict with the chosen units for HTB2 file I/O.

HTB2 uses a large number of unit identifiers, over 11, which may be difficult to allocate individually on some machines. Most of these identifiers,however, are not needed simultaneously, so that fewer that 11 units need to be used. The following table illustrates the usage of the unit identifiers in HTB2 in 'time', i.e. through the various sections of execution. The output units BK1UNT, BK2UNT, and LOGUNT , and the input units DY1UNT, DRYUNT, METUNT, will be used only if their appropriate routines (REPORT, PRFOUT, LOGGER , DIARY , METBAS respectively) are enabled. The zipper unit identifiers ZP1UNT, ZP2UNT are at present unused.



I/O UNIT USAGE

APPENDIX 3 - OUTPUT FILE FORMATS
Output File Formats

1.0 Output File Formats

This appendix describes the formats of the output data files produced by HTB2. These files, at the current release, are all sequential formatted files.

1.1 File Hierarchy

HTB2 output is organised into different files, corresponding to different data requirements. Each type is selectable by option, and the specific data in each type is also specifiable.

```
I           R           P           L
N           E           R           O
F           P           O           G
O           O           F           G
            R           I           E
            T           L           R
                        E
```

INFO	contains run information to help postprocessor programmes to interpret data in the other files (always required)
REPORT	contains interval averages of chosen fields of data.
PROFILE	contains instantaneous temperature and heat flux profiles of selected elements, at selected intervals
LOGGER	contains 'significant' changes to selected variables, data may be output at any time if value changes significantly. The significant level is selectable.

1.2 INFO output file

This file contains information about the run, stored for possible use by post-processing programmes, or for increasing the tracability of old runs. The name of this file is included in each of the other output files.

The information stored in this file includes;
the run time and system stamp,
data set identification strings,
data file names,
run option values,

data totals,
space and surface maps.

1.3 REPORT output file

The block interval report is sectioned into areas of interest:

METEOROLOGICAL
SPACES
WATER
VENTILATION
RADIANT ZONES
ELEMENT SURFACES
HEATING SYSTEM,

each of these sections is optional and the data in each represents the time average of the values over the report interval.

The file has a top section giving run date, id strings and a pointer to the appropriate INFO file.
Then a record containing the logical flags for the requested areas, written as;

spaces zones elements met heat water vent

A record of data totals follows;

number of spaces, number of zones, number of elements, number of heating systems

And finally the run timestep.

The report areas are written (if requested) in order;

met spaces water vent radiant element heating

MET BLOCK contains, in order:

AIR TEMP °C ,
WINDSPEED m/s ,
WIND DIRECTION o from N ,
HUMIDITY % ,
WATER CONTENT g/m3 ,
DEWPOINT oC,
CLOUD COVER 0-1,
DIFFUSE HORIZONTAL SOLAR W/m2,
DIRECT HORIZONTAL SOLAR W/m2,
GROUND TEMP °C ,
SUN ALTITUDE 0-90 o,
SUN AZIMUTH o cw from S.

SPACES BLOCK contains in order:

(virtual spaces only)	(or for modeled spaces)
SPACE num,	SPACE num,
AIRTEMP °C.	AIRTEMP oC ,

TOTAL CONVECTIVE GAIN W,
HEAT SYSTEMS CONVECTIVE GAIN W,
OCCUPANCY CONVECTIVE GAIN W,
SMALL POWER CONVECTIVE GAIN W,
OTHER SOURCES CONVECTIVE GAIN W,
FABRIC NET CONVECTIVE GAIN W,
VENTILATION CONVECTIVE GAIN W,
COOLING SYS CONVECTIVE GAIN W.

WATER BLOCK contains in order:
(virtual spaces only)

SPACE num,
HUMIDITY %,
WATER CONTENT g/m3,
DEWPOINT oC.

(or for modeled spaces)

SPACE num,
HUMIDITY %,
WATER CONTENT g/m3,
DEWPOINT oC,
WATER GAIN FROM HEAT SYSTEM g/s,
WATER GAIN FROM OCCUPANCY g/s,
WATER GAIN FROM FABRIC g/s,
WATER GAIN FROM SMALL POWER g/s,
WATER GAIN FROM OTHER SOURCES g/s,
WATER GAIN FROM VENTILATION g/s.

VENTILATION BLOCK contains in order:

SPACE num,
FRESH AIR CHANGE RATE /hr,
TOTAL AIR CHANGE RATE /hr.

RADIANT BLOCK contains in order:
(virtual zones only)

ZONE num,
MEAN RADIANT TEMP oC,
MEAN SURFACE TEMP oC,
TOTAL DIFFUSE RAD GAIN W.

(or for modeled zones)

ZONE num,
MEAN RADIANT TEMP oC,
MEAN SURFACE TEMP oC,
TOTAL DIFFUSE RAD GAIN W,
DIFFUSE GAIN FROM LIGHTING W,
DIFFUSE GAIN FROM OCCUPANTS W,
DIFFUSE GAIN FROM SOLAR W.

ELEMENT SURFACE BLOCK contains in order:

ELEMENT num,
INTERNAL SOURCE GAIN W/m2,

HEAT ABSORBED W/m2,
SOLAR TRANSMITTED W,
SOLAR ABSORBED W/m2,
LEFT/RIGHT SURFACE TEMP oC,
LEFT/RIGHT EFFECTIVE TEMP SEEN oC ,
LEFT/RIGHT HEAT FLUX W/m2, (+ve : inflow to material)
LEFT/RIGHT RADIANT GAIN W/m2,
LEFT/RIGHT RAD GAIN FROM HEAT SYSTEM W/m2,
LEFT/RIGHT RAD GAIN FROM SMALL POWER W/m2,
LEFT/RIGHT RAD GAIN FROM OTHER W/m2,
LEFT/RIGHT RAD GAIN FROM SOLAR W/m2,
LEFT/RIGHT CONDENSATION fraction of time below air dewpoint.

HEATING BLOCK contains in order:

SYSTEM num ,
SYSTEM DUTY CYCLE fraction time on,
SYSTEM OUTPUT W.

1.4 element PROFILE output

The PROFILE interval report produces temperature and heat flux section through the depth of selected elements.

The file has a top section giving run date, id strings
and a pointer to the appropriate INFO file.
A record of data totals follows
number of elements total, number of elements profiled
And the run timestep.

Preliminary data on the element at the file top is written as, for each element to be profiled:

ELEMENT num,
AREA m2,
SPACE TO LEFT/RIGHT,
ZONE TO LEFT/RIGHT,
CONSTRUCTION TYPE,
NUMBER OF PARTS,
NUMBER OF TEMPERATURE NODE,
NUMBER OF FLUX NODES.

Then

MATERIAL CODE,
NUMBER OF SLICES for each part,
FLUX NODE DEPTH (m), FOLLOWING TEMP NODE for each slice.

Interval data is output at the end of each interval and represents the instantaneous state at that time:

```
  ELEMENT number,
  NET HEAT ABSORBED W/m2,
  SOLAR POWER ABSORBED W/m2,
  SOLAR DIFFUSE TRANSMISSION W,
  SOLAR DIRECT TRANSMISSION W
  LEFT SURFACE:
      AIR TEMP oC,
      MRT oC,
      MEAN SURFACE TEMP oC,
      EFFECTIVE TEMP SEEN oC,
      RADAINT GAIN W/m2.

  TEMPERATURE of each slice oC, (left to right)
  HEAT FLUX of each slice W/m2, (left to right)

  RIGHT SURFACE:
      AIR TEMP oC,
      MRT oC,
      MEAN SURFACE TEMP oC,
      EFFECTIVE TEMP SEEN oC,
      RADAINT GAIN W/m2.
```

A temperature slice is at the center of each slice, and at each part boundary.
A flux slice is at each slice or part boundary.
Heat flux is +ve for heat flowing from left to right.

1.5 LOGGER output

The event logger outputs data on selected items only when they have changed in value by a significant amount from the last reported value.

LOGGING DATA is output as

```
  ITEM CODE ,
  DATE ,
  TIME ,
  INDEX (i.e. space, element number ) ,
  VALUE at last timestep,
  NEW VALUE.
```