UK REPRESENTATIVE BASIN AND FLOOD EVENT DATA

January 10th, 1992

Editors - David Boorman and Helen Houghton-Carr

Flood Event Modelling Section
Institute of Hydrology
Maclean Building
Crowmarsh Gifford
Wallingford
Oxfordshire

UK REPRESENTATIVE BASIN AND FLOOD EVENT DATA

January 10th, 1992

Editors - David Boorman and Helen Houghton-Carr

Flood Event Modelling Section
Institute of Hydrology
Maclean Building
Crowmarsh Gifford
Wallingford
Oxfordshire

List of contributors

The following members of the Flood Event Modelling Section have contributed to this document.

M C Acreman
N J Bonvoisin
D B Boorman
M C Clayton
B Gannon
H A Houghton-Carr
M Polarski

Table of Contents

1.0 Introduction

This document is a description of the data structures and methods of viewing, editting, retrieving and analysing data used for flood event modelling. As well as the time series data, and associated housekeeping information, there is a considerable volume of information describing sites at which the data are collected and the nature of the catchment areas draining to gauging stations. All of the data are stored on the IBM mainframe and use the ORACLE database management system.

Section 2 shows how tasks are performed using the various programs and execs that already exist. This is the place to start if you are a new user of the database and want to know how to do various jobs. For example, the user is led through the stages in defining a new event and preparing a catchment average rainfall. This section acts as an extended index to the descriptions of programs and EXECs (in Section 3), and ORACLE utilities (in Section 4).

There are three ways in which the data can be viewed, inserted, updated or deleted. These use different interfaces to the ORACLE data base: SQL*plus (a query language), SQL*forms (an interactive form) and a FORTRAN interface that allows data to be accessed directly from programs. Details of stored routines under each interface are given in Section 4. There is a fourth interface, the ORACLE data loader (ODL), which is designed for loading large data sets that already look like ORACLE tables. This is not used on a routine basis in the management of the flood event archive. The notes on the SQL*plus and SQL*forms may be of interest to casual users of the archive; only more serious users will be interested in the FORTRAN subroutine descriptions.

Section 5 describes the data structures and some of the basic links between them.

1.1 How to get a copy of this document

This manual is stored on the IBM mainframe on user-id FLOOD's D-disk in file DATABASE MANUAL. To get a copy, link to this disk and input the command

```
    scribe DATABASE MANUAL * device duplex
```

where * is the file mode (disk access letter) you have assigned to FLOOD's D-disk (this is D of course if you are logged on to FLOOD). See Section 2.1 for details of how to link to FLOOD's D-disk.

and `device` is either `TERM` to view the document in a formatted state on a terminal (remember not to use a 132 column terminal to do this), or `3812` to send it for printing on the IBM 3812 laser printer. Viewing at the terminal is only really appropriate to check the formatting of newly entered text.

and `duplex` sets the page formatting for single- or double-sided documents; it may be either `DOUBLE` or `SINGLE`.

1.2 Notes for contributors to the manual

Very few formatting commands need to be inserted in text to be entered in the manual as many defaults have been set up to define page layout etc.

- Headings need '<colon>hn.' at the front of the line, where n is the heading level. This section has an h2 title. All headings, down to h4, are taken and placed in the table of contents at the start of the manual. New pages and blank lines are automatically set up as appropriate to that level of heading.

- Each paragraph should be started with '<colon>p.'. Within a paragraph it is recommended that each new sentence is started on new line. The text will be shuffled around and justified when printed or displayed.
- Whenever a section should not be formatted in this way then type '<stop>fo off' before the part not to be formatted and '<stop>fo on' after it so that following sections will be formatted.
- Lists can be either ordered or unordered. In both types list items are identified with '<colon>li' causing a number or blob and the indentation of subsequent text. Ordered lists are started with '<colon>ol' and ended with '<colon>eol'; for unordered lists use '<colon>ul' and '<colon>eul'. These formatting notes are an unordered list.
- Default options set up fonts for all the text except tables that rely on spaces to maintain column positions. Before such a table put '<stop>bf tables' and after the table put '<stop>pf'. Such tables will normally also be unformatted text. Examples are also included in this manner.

1.3 Concepts

The original archive of flood event data was built up specifically for the Flood Studies. Events were stored in card image files that were easy to view but quite hard to edit and use for any other purpose. These files were known as PFILES (Processed data files) and until the latest version of the archive were still used as a way of passing data to analysis programs.

The present archive is more complex in that data are stored in many different places, but easier to use because each data structure is simple. The main difference between the ORACLE implementation of the flood event archive and earlier versions of it is that, whereas previously data were stored against events, there is now a clear distinction between data and the definition of the event. Data are stored in tables with names that fairly clearly identify their contents (eg. FLOW, RAIN, ESMD, CARAIN3). Only one flow value can be stored for a given gauge and time so where events overlap the data are stored just once. This separation of the data and event definition makes the data readily available for other applications, and allows many different types of events to be defined. Each data string required for an event is called a COMPONENT. So, for example, if an event is a sequence of flows and rainfalls then one component will identify the flow gauge number, the start and end times of the flow data, and the data interval; a second component will similarly specify the rainfall data. The two components are linked by a number that is unique to the event; these event numbers are allocated in the order events are created and while it is sometimes necessary to know it (eg. when creating a catchment average rainfall), in the programs needed for FSR analysis, events can usually be refered to by catchment number and start date.

At first glance the archive may appear complex, but the basic data structures, the ORACLE tables, are simple and so are the links between tables. Obviously care must be taken when changing or deleting data so remember, whenever possible, use the programs and execs that already exist to perform database operations. By using them you can be sure that the integrity of the database is maintained. Protection is also provided by only allowing restricted access to the data; users can be allocated read-only permission for example. The use of such restrictions is not only to protect the data but also to give user confidence by providing the assurance that they cannot change the data.

2.0 Methods

2.1 FSR unit hydrograph and losses model

In order to run FSR unit hydrograph and losses model analysis programs, data are required for flow, rainfall - both hourly and daily for the event and daily for the preceeding 5 days, and soil moisture deficit. These data are all stored on ORACLE tables.

A good place to start is to look at the data stored for a catchment. To do this you must be able to link to user-id FLOOD's D-disk and be able to log on to ORACLE. A password is needed to access FLOOD's D-disk, and may be obtained from David Boorman. You will need to put these commands in the appropriate place in your PROFILE EXEC A file.

```
'EXEC OLINK'

address command 'CP LINK FLOOD 192 296 RR ******'
'ACCESS 296 D'
```

The link number and access letter used are examples only, but you may find it less confusing if FLOOD's D-disk is also your D-disk.

Use CATCHEV to see what events exist. CATCHEV is a stored set of SQL*plus commands that will produce a neat lineprinter listing summary of the events. It is described in Section 4.2.2.1, but to use it just type

```
    START CATCHEV.SQLSP
```

and then input the catchment number.

There might also be additional flow stored for the catchment but not yet used. To obtain a list of dates for which flow values are available type

```
    START FLOWDATE.SQLST
```

and input the catchment number when requested.

Perhaps you have even more data that needs to be entered on the FLOW table, in which case use the

```
    LOADER LOADFLOW
```

program as detailed in Section 3.1.1.

The next type of data necessary is rainfall, hourly and daily. Obtain as much hourly data as possible to cover the event. Check what is on table RAIN already by running

```
    START RAINHOUR.SQLST
```

This will list all the dates for which we have some rainfall values, but bear in mind that this is not necessarily for the whole day. Additional rain can be put on by using the programs

```
    LOADER RAINLOAD

    LOADER HOURLOAD

    LOADER DRLOAD
```

The catchment must have a bounding quadrilateral with corners entered onto ORACLE table CATCHMENT_QUAD, and a polygon entry on table SEARCH_POLY (Section 3.2.4.2).

Then the daily rainfall for all the raingauges in or near the catchment can be transferred from Cache-Cache to ORACLE using

```
GETQRAIN
```

This takes a long time to run, so use it in batch (3.2.4.4).

Next we need to allocate event numbers by using the program

```
EVDEFS2
```

under the FLOOD user-id, see Section 4.3.2.

Now we can do something more interesting, the Catchment Average Rainfall Profile run by typing

```
CARP
```

Full details for running this program are given in Section 3.4. A large amount of output is generated and this needs to be studied and analysed. Sometimes it is necessary to re-run events for differing numbers of days to obtain the best results.

The next program in the sequence is

```
PERCH
```

for Plotting the Event Rainfall and Catchment Hydrograph. To start with it is a good idea to look at these plots on the screen before sending them to the Benson plotter or IBM 3812 laser printer. Full details are given in Section 3.5. From a study of the plots, decide whether the start and end times of flow or rainfall need to be changed, if so do this and re-run both CARP and PERCH.

The start date of the RAIN determines the date of SMD. The values from individual gauges are recorded on table ESMD. Missing values can be obtained from the Met. Office and entered using

```
LOADER ESMDLOAD
```

as described in Section 3.1.5. The Catchment Average SMD figure has to be calculated manually for each event and entered separately on table CASMD. When this has been done, the COMPONENT table will need an additional entry for the date of the Catchment Average SMD. This is also described in Section 3.1.5.

SO WHAT HAPPENS NEXT?

2.2 *Unit hydrograph model results*

Results from analysing events are stored on ORACLE tables so that between-event and between-catchment comparisons can be easily made. Results tables not only hold derived variables (eg. unit hydrograph time-to-peak and percentage runoff), but also key event parameters (eg. rainfall depth and peak flow). The following tables are all relevant to unit hydrograph and losses model analysis results

- FSR_RESULT: contains results from original FSR. Contains event start date, and has catchment, event_number AND event. To link to time series or other results use table EVLIST.

- FSSR16_RESULT: contains results as used in preparation of FSSR16. Some results are contained for events that are graded as unacceptable for UH analysis, or suitable for losses only. Only use the results in conjunction with table FSSR16_CODE. Results are stored for FSR events (but re-analysed and possibly different) as well as new events collected after the FSR was published. Contains event start date but should use start of flow data as in ORACLE table COMPONENT. Has catchment, event_number AND event.

- POST_FSSR16_RESULT: contains results derived since publication of FSSR16. Does not contain a date or catchment number; these are found by linking the event number to ORACLE table COMPONENT for data type 1 (ie. flow).

- FSSR16_CODE: contains code indicating quality of FSSR16 event data. (This table used to be called FSR_CODE but this was a bad name as the codes refer to FSSR16 not FSR).

- FSSR16_CATCHMENT_RESULT: contains catchment average values of parameters like Tp and PR. This needs to be a table as the calculation of SPR requires raising to non-integer power which could not be done with an SQL view. Marianne Polarski has a SQL*plus routine to derive averages. The values have been derived after checking quality codes in FSSR16_CODE.

There are also some views that relate to results

- UH_ANALYSIS_RESULT: this is a UNION of FSSR16_RESULT and POST_FSSR16_RESULT.
- NSHB_RESULT: this illustrates a way of of using a view to restrict data seen to just selected catchments.

3.0 Programs and EXECs

3.1 LOADER

LOADER is an EXEC that is used to load data from files to the appropriate ORACLE table. The EXEC and associated programs are held on FLOOD's D-disk. LOADER expects to be given a program name, a data file name, and an instruction as to whether output should be directed to the screen or the printer.

Note that with all of the loading programs a data item is NOT replaced if the new value is different from the old value. If an attempt is made to replace existing data a warning message will be printed for every value. To replace data, CAREFULLY delete existing data using SQL*plus, and then insert the new data.

Programs and corresponding data types are

1. LOADFLOW: for loading flow data.
2. HOURLOAD: for loading hourly rainfall digitised from Dines charts.
3. RAINLOAD: for loading rainfall typed into file.
4. DRLOAD: for loading daily rainfall not available from Met. Office tapes.
5. ESMDLOAD: for loading SMD gauge data (not catchment average values).

3.1.1 Loadflow (Table: FLOW)

Data can be entered into a file by hand, or can be the output file from digitising charts. Input the timing details using the normal 24-hour clock, remembering that midnight is 00.00 on the day following, and that the timing interval is given as 1.0 for 1 hour, 0.3 for half-hour, and 0.15 for quarter-hour.

3.1.1.1 Data format

```
1st line   3I5                  Catchment number, (event number,
                                data type)
                                Event number used for digitising only
                                and not loaded onto ORACLE.
                                Data type is 1 for flow.
2nd line   10x,I3,2x,3I2,2F5.2  Number of flow values, date, data
                                interval, start time as hh24.mi
3rd line+  Free                 Flow data in cumec
                                (must be same number of values as
                                specified on line 2)
NB. No blank lines.
```

3.1.1.2 Running job

Type LOADER LOADFLOW. The program logs on to ORACLE and asks for the name of the data file (eg. FLOW DATA D). The program assumes the data are in a variable format file but if records have a fixed length add an extra argument F (ie. FLOW DATA D F). The program then asks if output should be directed to the terminal or the printer (respond with TERM or PRINTER as appropriate).

3.1.1.3 Reformatting of flow data

A FORTRAN program called STAGEL will read hourly flow or stage values for a specified catchment, rate stage levels and write the data to file STAGEL OUTPUT A (fixed record length of 80 characters) in a format ready for LOADER LOADFLOW. STAGEL EXEC is also held on FLOOD's D-disk.

The format of input file STAGEL INPUT A (fixed record length of 80 characters) is as follows

```
1st line    I5,1x,A5              Catchment number, either 'STAGE'
                                  or 'FLOW'
2nd line    I3,2x,3I2,2x,F5.2     Number of stage or flow values,
                                  day, month,year, start time
3rd line+   Free                  Stage or flow data in m or cumec
                                  (must be same number of values as
                                  specified on line 2)
nth line    I5,1x,A5              As line 1, but must be for same
                                  catchment
```

The maximum number of values in line 2 is set at 240, if this is exceeded the program will stop with a warning message.

Unless 'FLOW' is specified in line 1, the stage values will be rated by the formula

```
    constant ( stage + alpha ) ** exponent
```

and rating equations (maximum of 10) must be supplied in input file STAGEL RATING A (fixed record length of 80 characters), one line per equation, with format as follows

```
1st line    Free                  Maximum stage height, constant,
                                  exponent, alpha (alpha may be negative)
2nd line+   Free                  Repeat up to total of 10 equations
```

The equations must be supplied in ascending order of stage height. If during the run a stage value is found to be greater than the greatest maximum stage height of the rating file the program will stop with a warning message.

Further programs can be developed to read data from other authorities as the need arises. See BG for details.

3.1.2 Hourload (Table: RAIN)

This program is for loading hourly rainfall data that have been digitised from Dines charts. Use program RAINLOAD if the data are to be typed into a file by hand (as it has an easier to enter data format).

3.1.2.1 Data format

All data to be in 24 hour blocks; the first value is the rainfall between 9:00 and 10:00.

```
1st line    I10,24x,I3,2x,3I2,2I4 Raingauge number, number of
                                  rainfalls, day, month, year,
                                  easting and northing
2nd line+   Free                  Hourly rainfall data in mm
                                  (must be the same number of values as
                                  specified on line 1)
```

3.1.2.2 Running job

Type LOADER HOURLOAD. The job will check if the gauge is known to ORACLE table RAIN_GAUGE_SITE, and if so whether the easting and northing agree with the values specified in the file.

3.1.3 Rainload (Table: RAIN)

This program is for loading rainfall data with any duration, provided this duration divides the day in an integer number of intervals. Input the timing details using the normal 24-hour clock, remembering that midnight is 00.00 on the day following, and that the timing interval is given as 1.0 for 1 hour, 0.3 for half-hour, and 0.15 for quarter-hour. Data must be loaded in complete rainfall days. This program is used if hourly rainfall data are typed into a file by hand as the data format is easier to cope with than that used for HOURLOAD. The file must be a variable format file.

3.1.3.1 Data format

1st line	Free	Raingauge number (0 or -1 to quit)
2nd line	I2,1x,F5.2	Number of days, data interval as hh24.mi
3rd line	I2,1x,I2,1x,I4	Day, month, year
4th line*	Free	Rainfall data in mm (must be the same number of values as calculated from line 2)
nth line	I2,1x,F5.2	As line 2 if more data for same gauge, 0 or -1 if not
mth line	Free	New gauge number if nth line 0 or -1

To check data prior to entry use program RAINFORM exactly as if loading the data. This program will list the start dates if everything is all right. The most common error is the wrong number of values.

3.1.3.2 Running job

Type LOADER RAINLOAD. The program will check to see if the gauge is known to ORACLE table RAIN_GAUGE_SITE. Do not be alarmed if the message gauge found but has different features is printed; this is a bug in the program but has no other effects.

3.1.4 DRload (Table: RAIN)

This program is only for loading data not on the British Rainfall archive, either because it is for a non-Met. Office approved gauge, or because it is very recent. If the data are on the daily rain archive then copy them using program GETQRAIN.

3.1.4.1 Data format

3.1.4.2 Running job

3.1.5 ESMDload (Table: ESMD)

This job is for loading SMD gauge data, not catchment average values. The SMD values are obtained from the Met. Office, usually by a personal visit to the section that keeps all the rainfall data. For information about whom to contact etc, please see the SMD folder kept at IH in Room 130.

For flood analysis purposes, we need to know the soil moisture deficit before the first rainfall of each event. The Met. Office calculate this value on a daily basis at some of their sites throughout the country, but the coverage is much more sparse than the raingauge network. Sometimes, therefore, the SMD stations may be quite a distance from the catchment concerned, and it may be necessary to get data from several stations in order to work out a catchment average.

For some unknown historical reason the daily SMD value is calculated by the Met. Office using the rainfall that has fallen during that day. This is apparent when looking at their tables of data; on the day that it rains, after a dry period when the value of SMD has been high, the SMD value will be lower. But what we need for our programs is the condition of the soil before the rain fell, so we have to take the SMD of the previous day. This value is shown on our database for the next day. Take care because this can create considerable confusion. Please remember, therefore, that if our event rainfall starts on 12 March 1980, we take the Met. Office SMD figure for 11 March 1980 but enter it onto our database as 12 March 1980.

Having obtained the necessary values for the dates required, and it is always a good idea to get more days than apparently needed, we can be load them onto ORACLE table ESMD using the LOADER ESMDLOAD program. For the SMD gauge number be sure to use the 'Hydrometric number' (usually the same as the raingauge number), and not the climate station number.

3.1.5.1 Data format

1st line	Free	Number of SMD gauges (x)
2nd line	Free	SMD gauge numbers
3rd line	Free	Number of days (y)
4th line	I2,1x,I2,1x,I2	Start date
5th line	Free	SMD values in mm (x per line for y lines)
nth line	Free	As line 3 if more days from the same gauges or -1 if not
mth line	Free	New number of gauges if more data available from a different set of gauges or -1 if not

3.1.5.2 Running job

Type `LOADER ESMDLOAD`. The program logs onto ORACLE and asks for the name of the data file (eg. SMD DATA A). The program assumes the data are in a variable format file. The program then asks if output should be directed to the terminal or the printer (respond with TERM or PRINTER as appropriate).

3.1.5.3 Catchment average SMD (CASMD)

If more than one SMD station is to be used, calculate the average SMD for the catchment for the rainfall start date using a simple or weighted average as appropriate. You will need to take into account the position of the gauge in relation to the catchment, its aspect, the type of soil, etc. If only one station value is available, then just use that figure.

Enter these values onto ORACLE table CASMD eg.

```
insert into FLOOD.CASMD (catchment, time, smd)
values (23006,to_date('&smddate 09.00','dd-mm-yy hh24.mi'),&smd);
```

Then enter the required details onto ORACLE table COMPONENT eg.

```
undefine smddate
insert into FLOOD.COMPONENT
(event, site, data_type, begin, until, interval)
values
(&event, 23006, 4, to_date('&&smddate 09.00','dd-mm-yy hh24.mi'),
                   to_date('&smddate 09.00','dd-mm-yy hh24.mi'),24);
```

Note that because only one day is involved, the same date has to be put in both the columns 'begin' and 'until'.

The routine SMDCOMPS.SQLI on FLOOD's D-disk inserts SMD components for a set of events with a particular collector/project code on a nominated catchment, and is convenient to use if you have more than a few events for the same catchment. The start date of the catchment average rainfall for each event is used to calculate the date of the SMD, so this job will not work unless you have run CARP (Section 3.4). In fact, inserting the SMD components is one of the last things you should do before running RUNLPROG (Section 3.6). To use, log on and links to FLOOD's D-disk, log on to ORACLE, and type

```
START SMDCOMPS.SQLI
```

You will be prompted for the catchment number, the collector/project code and the lowest and highest event numbers of those events requiring SMD components eg.

```
START SMDCOMPS.SQLI
Enter value for site:
23006
Enter value for collected_by:
13
Enter value for first_event:
4219
Enter value for last_event:
4223
```

```
5 records created.
```

The SMD components which have been loaded onto ORACLE can be checked by logging on to ORACLE and typing the SQL*plus command

```
SELECT * FROM COMPONENT WHERE EVENT=4219;
```

```
  EVENT DATA_TYPE    SITE BEGIN     UNTIL     INTERVAL
------- --------- ------- --------- --------- --------
   4219         1   23006 26-OCT-80 30-OCT-80        1
   4219         4   23006 26-OCT-80 26-OCT-80       24
   4219        10   13545 26-OCT-80 28-OCT-80       24
   4219        10   13553 26-OCT-80 .........       ..
   ....        ..  ...... ......... .........       ..
   ....        ..  ...... ......... .........       ..
   4219        10  604039 26-OCT-80 28-OCT-80       24
   4219        13   23006 26-OCT-80 28-OCT-80        1
   4219        17   23006 21-OCT-80 25-OCT-80       24
```

3.2 POLYON - Quadrilaterals and search polygons

3.2.1 Introduction

Before running the Unit Hydrograph and losses analysis programs on storm events on a catchment, some information associated with the catchment must be loaded onto the relevant tables in the ORACLE database.

3.2.2 Representing catchment boundaries by quadrilaterals

For the purposes of searching a catchment eg. for a list of available raingauges, the topographic boundary of most catchments can be adequately, though crudely, represented by a quadrilateral (Figure 3.2.1). Such a quadrilateral may be defined by the Ordnance Survey grid reference of its four corners, and these data are stored on ORACLE table CATCHMENT_QUAD. To check whether or not the quadrilateral for a catchment has been loaded, the following SQL*plus command is required

```
SELECT * FROM CATCHMENT_QUAD
        WHERE CATCHMENT=19001;

CATCHMENT CENTREX CENTREY   QGRID  QUADX1  QUADY1  QUADX2  QUADY2  QUADX3
--------- ------- ------- ------- ------- ------- ------- ------- -------
 QUADY3  QUADX4  QUADY4
------- ------- -------
    19001                       0    2825    6605    3048    6553    3195
   6710    3080    6785
```

If the quadrilateral for a catchment has not been loaded, it will be necessary to do so. The four corners must be put on in an anticlockwise order, with none of the x coordinates or y coordinates the same. eg.

```
INSERT INTO CATCHMENT_QUAD
      (CATCHMENT,QUADX1,QUADY1,QUADX2,QUADY2,QUADX3,QUADY3,QUADX4,QUADY4)
      VALUES
      (19001,2825,6605,3048,6553,3195,6710,3080,6785);

1 record created.
```

3.2.3 Defining search polygons

To search the area enclosed by the quadrilateral, each point on its boundary must be defined. This can be achieved by representing each of the four boundary lines by an equation. Each line can be uniquely defined by a slope and constant, together with the minimum and maximum values. This is achieved using an SQL*plus subroutine called POLYON. When running POLYON the search area can be easily increased or decreased by specifying an expansion factor together with the catchment number. In SQL*plus type

```
START POLYON.SQL
Enter value for factor:
1.5
Enter value for catchment_number:
19001

1 record created.

1 record created.

1 record created.

1 record created.
```

This routine retrieves the location of the four corners from CATCHMENT_QUAD, calculates the equation for each line and makes four entries onto ORACLE table SEARCH_POLY. To check this, or to see if POLYON has already been run for a catchment, the following SQL*plus command is required

```
SELECT * FROM SEARCH_POLY
```

```
        WHERE PID=19001;

    PID     MINY     MAXY    SLOPE CONSTANT   FACTOR
------- -------- -------- -------- -------- -------
  19001     6498     6576  -4.2885   7209.9      1.5
  19001     6498     6733   .93631  3236.65      1.5
  19001     6733     6846  -1.5333  8868.59      1.5
  19001     6576     6846  1.41667  4656.58      1.5
```

3.3 *GETQRAIN - Loading daily rainfall data from Cache-Cache*

3.3.1 Introduction

The Institute of Hydrology maintains an archive of daily rainfall data from storage gauges throughout the UK for the period 1961-1989 supplied by the Met. Office. Since this comprises a very extensive volume of data it is stored on a separate database called Cache-Cache. Before running the Unit Hydrograph losses and analysis programs on storm events on a catchment, it is necessary to transfer daily rainfall data for relevant raingauges from Cache-Cache to ORACLE table RAIN using the program GETQRAIN.

The program searches 10km Ordnance Survey grid squares for the expanded catchment area and transfers all available daily rainfall data from Cache-Cache to ORACLE table RAIN. An entry is also made on ORACLE table SEARCH_RAIN_SQUARE_DATE, thus indexing the loading of daily rainfall by grid square and date.

3.3.2 GETQRAIN Pre-requisites

In order to run GETQRAIN successfully, some data associated with the catchment must have been loaded onto the relevant tables in the ORACLE database.

- RIVER NAME AND LOCATION - Each gauging station in the UK, which is recognised by the Surface Water Archive, is given a catchment number. This number is stored against the names of the station and the water course on ORACLE table SWA.STATION_NAME. GETQRAIN cannot be run on catchments which are not recognised by the Surface Water Archive.

- QUADRILATERAL AND SEARCH POLYGON - GETQRAIN checks that the location of each of the four corners of the catchment has been loaded into CATCHMENT_QUAD, and that the boundary information has been loaded into SEARCH_POLY. A fatal error results if the relevant information is not found. See Section 3.2 for details.

- CHECKING PREVIOUSLY LOADED DATA - this is not mandatory but may save time later when running GETQRAIN. Whether data has been loaded onto ORACLE for the required grid squares and periods can be checked by interrogating ORACLE table SEARCH_RAIN_SQUARE_DATE where each grid square is uniquely defined by the grid reference of its south-western corner. Eg for a catchment in grid square 30 67 and a period from 2nd September to 5th September 1978, the following SQL*plus command is required

```
SELECT * FROM SEARCH_RAIN_SQUARE_DATE
        WHERE EAST=30 AND NORTH=67
        AND DAY BETWEEN TO_DATE('02-SEP-1978','DD-MON-YYYY')
                    AND TO_DATE('05-SEP-1978','DD-MON-YYYY');
```

```
   EAST   NORTH DAY       LOADED
------- ------- --------- ---------
     30      67 02-SEP-78 25-APR-88
     30      67 03-SEP-78 25-APR-88
     30      67 04-SEP-78 25-APR-88
     30      67 05-SEP-78 25-APR-88
```

  Alternatively, where the grid squares are not known, the very slow SQL*plus retrieval WHATRAIN can be run to check whether daily rainfall has been transferred to ORACLE table RAIN for a catchment during a previous run. See Section 4.2.2.3 for details.

- File GQR DATA A1 - specifying the catchment reference number, an expansion factor, data acceptance code (0 or 1) and the dates of rainfall to be loaded. The file must have the following format

```
1st line        Free              Catchment number
2nd line        F5.2              Quad enlargement factor
3rd line        I1                Data acceptance code 0 or 1 (see below)
4th line        2I2,I4,2I3        Day, month, year, length of event
                                  in days, API in days
5th and subsequent lines as line 4
```

If rainfall for other catchments is required, type 0 on two consecutive lines after the last event of the previous catchment and repeat from line 1 onwards. End the file with 0 on three consecutive lines. For example

```
19001
1.5
1
01011978  4  6
12121979  3  6
15111980  7  6
0
0
19002
1.5
1
01011978  4  6
0
0
0
```

Always ask for one more day than you think is necessary in case starting dates are changed for any reason.

Data acceptance code set to 0 means to exclude estimated rainfall from the transfer, 1 means to include it. In the latter case, estimated rainfall will be held as a negative value on table RAIN (this facility has been available for data loaded after May 1990, before that date estimated rainfall was not transferred; see BG for a program to 'reload' estimated rainfall not transferred previously). Some rainfall data has accidentally been loaded for gauges in Northern Ireland and the Channel Islands. It is therefore essential to include the grid or zone code when matching raingauges with catchments before processing rainfall data (no more data from these gauges is expected to be transferred from July 1990, and all relevant analytical programs have a zone check in their subroutines).

3.3.3 Running GETQRAIN

The program takes a considerable length of time to run and should be submitted to either VMSchedule (overnight) or VMBatch. VMSchedule was originally designed to run under user-id FLOOD but can be run under other user-ids provided disk allocations do not clash. VMBatch cannot be run under user-id FLOOD because of clashing disk allocations.

To submit the program to VMSchedule, type

```
VMSCHED SCHED myjob BATGQR (AT 22:00
```

Myjob is a unique 8-character name to refer to the VMSchedule request and should include the user's ID. You will then be asked to input your logon password.

To run the program under VMBatch, establish availability of Class by typing

```
VMBATCH Q QUEUE
```

and choose between Classes Q (20 minutes CPU time maximum) or O (24 hours CPU time maximum). Then type

```
VMBATCH SUBMIT GETQRAIN (CL Q TIME 20:00 NAME name
```

or

```
VMBATCH SUBMIT GETQRAIN (CL O TIME 24:00:00 NAME name
```

Name is a 6-character name to identify the batch job and should include the user's ID.

3.3.4 Output

For each event, GETQRAIN produces a list of raingauges in grid square order for which daily rainfall has been loaded during this run onto ORACLE table for the relevant period. A second summary is printed showing the total number of raingauges and rainfall data availability by grid square.

If rainfall had already been transferred during previous runs of GETQRAIN for the same grid square and date combinations, then the comment `All data loaded` is printed instead and the program does not overwrite previously loaded rainfall.

3.3.5 GETQRAIN subroutines

GETQRAIN uses several of the FORTRAN subroutines described in Section 4.4 to perform various database operations. These include

- CZONE - used to retrieve the catchment zone code from ORACLE table SWA.STATION_DETAIL.
- QUOFF - used to retrieve the corners of the catchment quadrilateral from ORACLE table CATCHMENT_QUAD.
- RAINON - used to load rainfall data onto ORACLE table RAIN.
- RGC10K - used to retrieve a list of raingauges for a 10km square from ORACLE table RAIN_GAUGE_SITE. The grid or zone code must be specified.
- SRSZOF - used to see if a particular 10km square reference and date are on ORACLE table SEARCH_RAIN_SQUARE_DATE. The grid or zone code must be specified.
- SRSZON - used to load a particular 10km square reference and date onto ORACLE table SEARCH_RAIN_SQUARE_DATE. The grid or zone code must be specified.

3.4 CARP - Catchment Average Rainfall Package

3.4.1 Forward

In many aspects of hydrological modelling, such as the transformation of rainfall into runoff, it is often sufficient to assume that rainfall is evenly distributed across a catchment. However, rainfall data is at present only widely available for point locations, thus these data need to be averaged over the catchment. The conventional method of achieving this is simply to average all the individual rainfalls at each time step, yielding an areal catchment average profile. An alternative conceptualisation is the catchment average point profile (Jones, 1983) which attempts to find an average shape for the profiles independent of timing. To achieve this the individual profiles must be aligned. Although this is relatively straight forward by hand, computerised versions of the procedure have often produced unrealistic average profiles, so this method has not been adopted for current use. This section describes a computer based package which derives an areal catchment average rainfall profile.

3.4.2 Introduction

CARP (Catchment Average Rainfall Package) comprises of a set of programs which calculate a catchment average rainfall profile, with an hourly time step, for an isolated event of up to 10 days duration. This is achieved by averaging all available rainfall data for the duration of the event. This will include daily data supplied by the Met. Office and previously transferred from the Cache-Cache archive by the program GETQRAIN, and hourly data obtained from recording raingauges for specific events. A subroutine searches the database for relevant rainfall data for the specified period of the event. The search is usually confined to the catchment itself and an area immediately surrounding it. Further subroutines average the daily rainfalls to give a catchment average event total. This is then distributed between the hours of the event, using an average profile calculated from the recording gauge data, to give the catchment average rainfall profile. Data from daily gauges for the five days previous to the event are also analysed to derive a catchment average antecedent precipitation index. Details of the calculations, the catchment average rainfall profile and the antecedent precipitation index are printed on the line-printer and the resulting data may be loaded onto the appropriate data archive, if required. An option is available to produce ROACH (Rainfall Over Area of Catchment Hyetographs and map) plots which provide an A4 size pictorial summary of the event, comprising a catchment map with the quad, relevant raingauges, and hourly and catchment average rainfall hyetographs. These maps can either be displayed on the screen, or sent to a plotter/printer for output on paper.

NB. CARP is not suitable for using on events where precipitation has fallen as snow, unless it has melted immediately.

3.4.3 Quick guide

1. Log on and link to FLOOD's D-disk
2. Type CARP and press <ENTER>
3. Wait for program to load
4. If the catchment map option is chosen, input the required type of output device. The choice is between
   - the SCREEN,
   - the BENSON plotter,
   - the IBM 3812 LASER printer A4 size (3812A4).
5. Input the catchment number. Either accept the one shown by pressing <ENTER>, or type a new one in integer format (up to 6 digits) and then press <ENTER>. Type a negative number to exit from the program. If you make a mistake, there will be a chance to correct it later.
6. Input the event number, if known. Either accept the one shown (00000) by pressing <ENTER>, or type a new one in integer format (up to 5 digits) and then press <ENTER>.
7. Input the start date of the rainfall. Either accept the one shown by pressing <ENTER>, or type a new one in the standard ORACLE format DD-MON-YYYY and then press <ENTER>. There is no need to input times at this stage.

8. Input the end date of the rainfall in the same way. The maximum number of days for an event is 10.

9. Input the number of days API. Either accept the one shown (5 days) by pressing <ENTER>, or type a new one in integer format and then press <ENTER>. The maximum number of days API is 10.

10. Input the number of the first extra gauge to be included in I6 format and press <ENTER>. If no extra gauges are required just press <ENTER>; gauge 898567 will not be included. Input additional gauges in the same way. The maximum number of extra gauges is 3.

11. The screen displays the input details for checking. Type Y and press <ENTER> if all the entries are correct, and go to step 12. If changes are required type N and press <ENTER>, and answer all the questions again. Previous entries will remain until changed.

12. If the event under analysis overlaps previous events the resulting data will not be stored, and the event should be re-run for the complete period covering both events.

13. If the data are to be stored, another input form appears on the screen. Input the start and end dates and times of the catchment average rainfall in the standard ORACLE format DD-MON-YYYY HH24.MI in the same way as before. Again the input details are available for checking.

14. If you have selected the catchment map option and are plotting on the screen, the screen will clear and the plot will appear after a few moments. Press <ENTER> to continue.

15. If the input period covers more than one event the start date may not indicate the start of the event of interest. If the data are to be stored, another input form may appear on the screen. Input the start date and time for the API calculations in the standard ORACLE format DD-MON-YYYY HH24.MI in the same way as before. Again the input details are available for checking.

16. To exit from the program select a catchment number of -1. If plots have been displayed on the screen MORE.. will appear in the bottom right of the screen. Press <CLEAR> and then <ENTER> to terminate the program.

17. If you are sending plots to the BENSON continue entering event details until finished, then exit from the program. Next type BENPLOT and press <ENTER>. Input any special messages for the operator and press <ENTER>. The plots will eventually arrive in your pigeon-hole.

18. If you are sending plots to the IBM 3812 LASER printer continue entering event details until finished, then exit from the program. Each event will be in your filelist as a separate file of the type I3812Pnn LISTAPA A where nn refers to a page number. Each file must be sent to the printer by the command PRT3812 fn ft fm (ORIENT E. Details of how to utilise the CMS command FILELIST to do this may be found in the February 1990 Wallingford User Note. The plots will soon arrive in your pigeon-hole.

19. Print-outs detailing the analysis of the rainfall will arrive in your pigeon-hole. Print-outs detailing the data entered onto ORACLE tables can be obtained by typing PRINT CLOAD LISTING A. Output will be sent to your pigeon-hole.

3.4.4 CARP Pre-requisites

In order to produce a successful catchment average rainfall profile, some data associated with the catchment for the storm event must have been loaded onto the relevant tables in the ORACLE database.

- RIVER NAME AND LOCATION - Each gauging station in the UK, which is recognised by the Surface Water Archive, is given a catchment number. This number is stored against the names of the station and the water course on ORACLE table SWA.STATION_NAME. The CARP program cannot be run on catchments which are not recognised by the Surface Water Archive.

- QUADRILATERAL AND SEARCH POLYGON - CARP checks that the location of each of the four corners of the.catchment has been loaded into CATCHMENT_QUAD, and that the boundary information has been loaded into SEARCH_POLY. A fatal error results if the relevant information is not found.

- AREA - The drainage area of the catchment is required. Values of the drainage area for all catchments recognised by the Surface Water Archive are stored on ORACLE table SWA.STATION_DETAIL.

- AVERAGE ANNUAL RAINFALL - for the standard period 1941-1970. Average annual rainfalls for all catchments recognised by the Surface Water Archive are stored on ORACLE table SWA.CATCHMENT_CLIMATE.
- HOURLY DATA - Data at an hourly interval from at least one recording raingauge are required to run CARP. These data must be obtained independently for each storm event from the relevant measuring authority, since IH does not hold a national archive.

3.4.5 Running CARP

CARP is run by typing CARP when linked to user-id FLOOD on the IBM.

```
THANKYOU FOR CHOOSING CARP
 PLEASE WAIT WHILE LOADING
```

appears on the screen. It takes a few moments to link the compiled subroutines together. This is followed by

```
DO YOU WANT ROACH PLOTS? Y/N
```

referring to the plots of the rainfall hyetographs and catchment map. If you reply Y, you will be prompted for the required type of output device

```
DEVICE TYPE   (A6)
  BENSON SCREEN or 3812A4
```

which you should input in the manner shown.

3.4.5.1 Form filling

Input of information to CARP is achieved by answering questions which appear on the screen. The questions have the following form

```
   CCCC       A      RRRR     PPPP
  C          A A     R   R    P   P
  C         A   A    R   R    P   P
  C         A   A    R  R     P  P
  C         AAAAA    RRR      PPP
  C         A   A    R  R     P
  C         A   A    R  R     P
  C         A   A    R  R     P
   CCCC     A   A    R  R     P

CATCHMENT AVERAGE RAINFALL PACKAGE

CATCHMENT NO -ve number to quit          19001
Press <ENTER> to accept or input new CATCHMENT NO
?

EVENT NO                                     0
Press <ENTER> to accept or input new EVENT NO
?
128

CARP INPUT DATA ENTRY (contd)

START DATE (A11)                         13-AUG-1966 09.00
Press <ENTER> to accept or input new START DATE

END DATE (A11)                           15-AUG-1966 08.00
Press <ENTER> to accept or input new END DATE
14-AUG-1966

NO DAYS API maximum 10 days                5
Press <ENTER> to accept or input new NO DAYS API
?
```

```
   CARP INPUT DATA ENTRY (contd)

   EXTRA GAUGES (16)                             898567
   Press <ENTER> for none or input EXTRA GAUGE NO
   899567
   Press <ENTER> for no more or input EXTRA GAUGE NO

YOU HAVE SELECTED DATA FOR CATCHMENT NUMBER 19001 FOR EVENT NUMBER 128
FOR THE PERIOD STARTING 13-AUG-1966 09.00 AND ENDING 14-AUG-1966 08.00
                   WITH 5 DAYS API AND 1 EXTRA GAUGE(S)
                       899567

 ALL CORRECT?  (Y/N)
```

- DEVICE TYPE - ROACH plots can be displayed on the terminal by selecting SCREEN, sent to the BENSON plotter by typing BENSON, or sent to the IBM 3812 LASER printer by typing 3812A4. It is not possible to swap between devices during a single run, though when plotting on the screen the data are stored in a file which can be sent to the BENSON later; note however that it is not possible to plot a subset of the events displayed on the screen as the file contains data from the entire CARP run. If you are sending plots to the IBM 3812 LASER printer there may be disk space restrictions as each event requires around 100 blocks, so make sure that you have sufficient room before you start. This may be achieved by sending large files not in current usage to your reader (by typing `SENDF fn ft fm TO *`) and erasing the originals from your filelist. There are various advantages and disadvantages of the two forms of hardcopy output: the BENSON style ROACH is in colour, but takes at least a day to appear and then requires trimming, whilst the IBM 3812 style ROACH is in black and white, but appears virtually instantaneously on A4 paper.

- CATCHMENT NO - This is the Surface Water Archive reference number given to each gauging station in the UK.

- EVENT NO - This is the reference number for the event which is stored on ORACLE table EVENT and is used to cross reference the various components which make up the event, including the flow rainfall and SMD data. This number is allocated when the component associated with the flow data for the event is set up (see Section 4.3.2 EVDEFS2 for details). If the resulting catchment average rainfall data are not to be stored on the database, an event number is not required. If the event number is not known or has not been allocated, 0 should be entered.

- START DATE - This is the date at 9.00am of the first day of the rainfall data for the event. The date must be entered in the standard ORACLE format DD-MON-YYYY. There is a check that the date is valid. If an EVENT NO has been given, the start date of the flow data associated with the event is used as an estimate of the start date of the rainfall.

- END DATE - This is the date at 8.59am of the last day of the rainfall data for the event. The date must be entered in the same format as the START DATE. Again there is a check that the date is valid. If available, the end date of the flow data is used as an estimate of the end date of the rainfall data. The maximum number of days for an event is 10.

- NO DAYS API - This is the number of days prior to the start of the storm event which are to be used to calculate an Antecedent Precipitation Index. The usual period is 5 days, and the maximum number of days is 10.

- EXTRA GAUGES - Raingauges which may not be found during the search of the expanded quadrilateral can be specifically included by entering their reference numbers here. Up to 3 additional gauges may be included. Type each number and press <ENTER>. If no extra gauges are required just press <ENTER>; gauge 898567 will not be included.

3.4.5.2 Components

Details of the various components associated with an event are found on ORACLE table COMPONENT. The component entries include details of flow gauges, daily and recording rain gauges, the catchment average rainfall profile, and the Antecedent Precipitation Index and the Soil Moisture Deficit, and comprise event number, data type, gauge number, start and end dates and times, and data interval in hours.

If an event number other than 0000 is entered, the program will search the COMPONENT table for information associated with this number, and any components detailing flow data and/or catchment average rainfall data previously produced by CARP are printed out eg. event number 128 has the following component details for the flow data and CARP catchment average rainfall data

```
  EVENT DATA_TYPE     SITE BEGIN     UNTIL     INTERVAL
------- --------- ------- --------- --------- --------
    128         1   19001 13-AUG-66 15-AUG-66        1
    128        13   19001 13-AUG-66 14-AUG-66        1
```

This provides an opportunity for checking that the correct event number has been chosen. If the input event number is 0000 no selections can be made.

If data storage is requested during the CARP run, details of the various components used to calculate the catchment average rainfall profile are stored in the COMPONENT table. The data which have been loaded onto ORACLE following a CARP run can be checked by logging on to ORACLE and typing the SQL*plus command

```
SELECT * FROM COMPONENT WHERE EVENT=128;

  EVENT DATA_TYPE     SITE BEGIN     UNTIL     INTERVAL
------- --------- ------- --------- --------- --------
    128         1   19001 13-AUG-66 15-AUG-66        1
    128         2   19001 13-AUG-66 14-AUG-66        1
    128         3   19001 13-AUG-66 14-AUG-66        1
    128         4   19001 13-AUG-66 13-AUG-66       24
    128         5  900093 13-AUG-66 13-AUG-66       24
    128         6   19001 16-JUL-66 13-AUG-66       24
    128         7   19001 08-AUG-66 13-AUG-66       24
    128        10  656041 13-AUG-66 13-AUG-66       24
    128        10  897213 13-AUG-66 .........       ..
    ...        ..  ......  ......... .........       ..
    ...        ..  ......  ......... .........       ..
    128        10  900117 13-AUG-66 13-AUG-66       24
    128        13   19001 13-AUG-66 14-AUG-66        1
    128        17   19001 08-AUG-66 12-AUG-66       24
```

Print-outs detailing this data can also be obtained by typing `PRINT CLOAD LISTING A` at the end of the CARP run. Output will be sent to your pigeon-hole.

3.4.5.3 Overlapping events

The same catchment average rainfall data (or in fact any other data type) may be associated with several different event definitions if the dates and times specified in the component entries overlap. For example a long multi-peaked storm event may be analysed firstly as one single event, and then as several events with each burst processed separately. Therefore, data recalculated with respect to a single burst will alter part of the data associated with the entire event. To ensure compatibility of all data within any event, analysis should always cover whole events. CARP searches the component entries for details of catchment average rainfalls which overlap the period specified for the current analysis. For each overlapping event found, one of the following messages is displayed depending on the relative timing of the events

```
1          COMPONENT FOUND IS WITHIN DATES GIVEN
2          COMPONENT FOUND ENCLOSES DATES GIVEN
3          COMPONENT FOUND OVERLAPS END DATE GIVEN
4          COMPONENT FOUND OVERLAPS START DATE GIVEN
5          COMPONENT FOUND WITH SAME DATES
```

If the event found has the same dates, or is enclosed within the current period, its entire data will be recalculated. If any component found satisfies one of the other descriptions, the period under current analysis should be extended to cover the whole of both events. In this case the program will not load any data or component entries onto ORACLE tables.

3.4.5.4 Deriving the catchment average rainfall profile

A list of all raingauges recognised by the Met. Office is contained on ORACLE table RAIN_GAUGE_SITE. Those gauges which are located within the search polygon of the catchment are listed as potential sources of data. Some of these will be daily storage gauges, and some will be recording gauges. The number of gauges found appears on the print-out. The maximum number that can be examined by the program is 2000. All rainfall data for these gauges, complete between the start and end dates, that are stored in the ORACLE table RAIN, are retrieved. The maximum number of gauges with data that can be manipulated is 100 daily and 10 hourly. If more than that number are available, the first 100 (10) with data are selected. A list of gauges with rainfall data is printed (see Figure 3.4.1). The gauges are divided into those for which daily totals are available, and those for which hourly data are available. If data are available at other intervals, a message is printed out.

Each daily rainfall gauge is given a weight according to its location within the catchment. Weights are derived by the method described by Jones (1983). The total event rainfall at each gauge is expressed as a proportion of the annual rainfall by dividing it by the at-site estimate of SAAR (1941-1970). Each standardised rainfall is multiplied by its weighting factor to yield a catchment average standardised event rainfall. This value is then rescaled by multiplying it by the catchment average SAAR, to obtain the catchment average total rainfall for the event. The data used in these calculations appear on the print-out (see Figure 3.4.2). Acreman (1986) compared estimates of point rainfalls using gauge totals standardised in this way, with those using the original observed totals. In general, during frontal storms, rainfall depths tend to exhibit a spatial distribution somewhat similar to that of the average annual rainfalls ie. event depths are higher where SAAR is higher. In this situation averaging the standardised rainfalls gives an improved catchment average. However, during cyclonic storms the rainfall depths tend to be more randomly distributed and bear little relation to the distribution of SAAR. Therefore estimates of the catchment average event rainfall may be better estimated by using the original gauge totals. Cyclonic rainfalls tend only to cause significant flood events on small catchment, so, on balance, using the standardised rainfalls is to be preferred. The output from CARP contains the correlation coefficient indicating the degree of dependence between gauge event totals and SAAR. When the correlation is high, near 1.0, using division by SAAR improves the estimate of the catchment average; when it is low the estimate is likely to be to be poorer.

A weight is derived by the same method as above for each rainfall gauge from which hourly totals are available. For each gauge, each hour is expressed as a proportion of the total event rainfall at that gauge. For each hour in turn, the proportion at each gauge is then multiplied by the gauge weight, and these weighted proportions are summed across all the gauges to yield a catchment average rainfall profile. The data used in these calculations appear on the print-out (see Figure 3.4.3) together with a graphical representation of the individual and catchment average profile shapes (see Figure 3.4.4). The geographical location of all the daily and hourly raingauges used is also displayed on the print-out (see Figure 3.4.5).

If no overlapping event component entries are found for the catchment average rainfall, the profile resulting from running CARP may be stored on ORACLE table CARAIN3. When the calculations are complete the user is asked if the data are to be stored.

```
DO YOU WANT TO LOAD UP THE
COMPONENT ENTRIES FOR THIS
EVENT? (Y/N)
Y

         CCCC      A      RRRR     PPPP
        C         A A     R   R    P   P
        C        A   A    R   R    P   P
        C        A   A    R  R     P  P
        C        AAAAA    RRR      PPP
        C        A   A    R  R     P
        C        A   A    R  R     P
        C        A   A    R  R     P
         CCCC    A   A    R  R     P

      CATCHMENT AVERAGE RAINFALL PACKAGE

      CATCHMENT NO                          19001
      EVENT NO                                128

      CARAIN3 COMPONENT START DATE          13-AUG-1966 11.00
      Press <ENTER> to accept or input new START DATE

      CARAIN3 COMPONENT END DATE            14-AUG-1966 08.00
      Press <ENTER> to accept or input new END DATE

   YOU HAVE SELECTED DATA FOR CATCHMENT NUMBER 19001 FOR EVENT NUMBER 128
   FOR THE PERIOD STARTING 13-AUG-1966 11.00 AND ENDING 14-AUG-1966 08.00

    ALL CORRECT?  (Y/N)
```

Catchment average hourly rainfall data resulting from running CARP are stored on ORACLE table CARAIN3. The data which has been loaded for a particular catchment and for a particular time period can be examined using the following SQL*plus command

```
SELECT TO_CHAR(TIME,'DD-MON-YY HH24.MI'), RAIN
  FROM CARAIN3
  WHERE CATCHMENT = 19001
  AND TIME BETWEEN TO_DATE('13-AUG-1966','DD-MON-YYYY')
                AND TO_DATE('14-AUG-1966','DD-MON-YYYY');
  ORDER BY TIME

TIME                    RAIN
------------------- -------
13-AUG-66 09.00            0
13-AUG-66 09.00         43.3
13-AUG-66 10.00            0
13-AUG-66 11.00            0
13-AUG-66 12.00           .2
13-AUG-66 13.00           .6
13-AUG-66 14.00          1.1
13-AUG-66 15.00          1.5
13-AUG-66 16.00            1
13-AUG-66 17.00          2.6
13-AUG-66 18.00          4.1
13-AUG-66 19.00            3
13-AUG-66 20.00          3.3
13-AUG-66 21.00          2.9
13-AUG-66 22.00          4.6
13-AUG-66 23.00          3.5
14-AUG-66 00.00          3.5
```

3.4.5.5 ROACH plots

If you want to plot on the screen you must be logged on to a 3179 graphics terminal. Enter SCREEN when DEVICE TYPE is asked for. When all the input information is correct, press Y, and the screen will clear and the plot will appear after a few moments. Press < ENTER > to continue. Once the plots have been examined on the screen they can be sent to the BENSON plotter without re-running CARP. Exit from the program by selecting a catchment number of -1, and when command has returned to CMS type BENPLOT and press < ENTER >. Input any special messages for the operators and press < ENTER >. The plotting information will be sent to the plotter, and the plots will eventually arrive in your pigeon-hole.

If you want to send your plots to the BENSON, without first displaying them on the screen, enter BENSON when DEVICE TYPE is asked for. When all the input information is correct, press Y, and you will then be prompted for details of the next event. Continue entering events details until finished, then exit from the program by selecting a catchment number of -1. The information for the plotter will be held in file B1645 PLOTCODE A. When command has returned to CMS type BENPLOT and press < ENTER >. Input any special messages for the operators and press < ENTER >. The plotting information will be sent to the plotter, and the plots will eventually arrive in your pigeon-hole.

If you want to send your plots to the IBM 3812, input 3812A4 when DEVICE TYPE is asked for. When all the input information is correct, press Y. On the screen will appear the words

```
Page size is A4 Landscape
IBM 3812 page printer - page nn
```

where nn refers to a page number ie. your first event will be page number 1, and so on. You will then be prompted for details of the next event. Continue entering event details until finished, then exit from the program by selecting a catchment number of -1. The information for the printer will be held in a separate file for each event, of the type I3812Pnn LISTAPA A where nn refers to the page number. When command has returned to CMS type PRT3812 fn ft fm (ORIENT E for each file. Details of how to utilise the CMS command FILELIST to do this may be found in the February 1990 Wallingford User Note. The files will be sent to the printer, and the plots will soon arrive in your pigeon-hole.

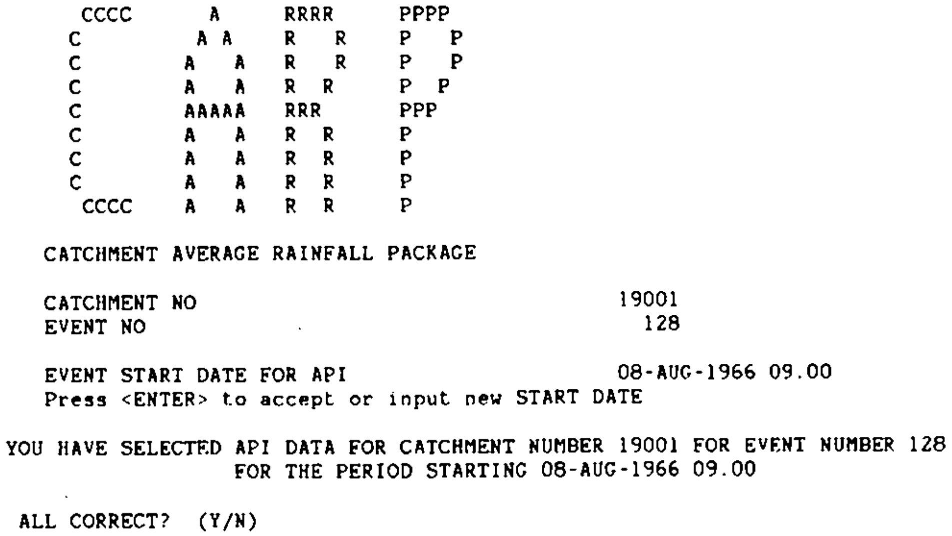
An example ROACH plot is illustrated in Figure 3.4.6.

3.4.5.6 Antecedent Precipitation Index (API)

The API provides an indication of the hydrological condition of the catchment prior to the event. The Flood Studies Report (NERC, 1975) recommends the use of a short-term exponentially decaying API based on the 5 previous days ie. API5.

The derivation of this index utilises the list of raingauges which are located within the search polygon and the number of days of antecedent rainfall to be analysed. All daily rainfall data for these gauges, complete for the antecedent period, that are stored in the ORACLE table RAIN, are retrieved. Each daily rainfall gauge is given a weight according to its location within the catchment. Weights are derived by the method described by Jones (1983). The data used in these calculations is given on the print-out (see Figure 3.4.7).

If the storm event under analysis overlaps another, CARP should be re-run for the whole period which encompasses both events. In this case the start date given will not necessarily indicate the start of the event of interest, therefore it is not appropriate to calculate the API relative to this date. When this situation occurs another form is presented on the screen which allows the start date to be changed to that which relates to the event of interest.



```
          CCCC       A      RRRR      PPPP
         C          A A     R   R     P   P
         C         A   A    R   R     P   P
         C         A   A    R  R      P  P
         C         AAAAA    RRR       PPP
         C         A   A    R  R      P
         C         A   A    R  R      P
         C         A   A    R  R      P
          CCCC     A   A    R  R      P

       CATCHMENT AVERAGE RAINFALL PACKAGE

       CATCHMENT NO                              19001
       EVENT NO                                    128

       EVENT START DATE FOR API                  08-AUG-1966 09.00
       Press <ENTER> to accept or input new START DATE

    YOU HAVE SELECTED API DATA FOR CATCHMENT NUMBER 19001 FOR EVENT NUMBER 128
                        FOR THE PERIOD STARTING 08-AUG-1966 09.00

     ALL CORRECT?  (Y/N)
```

If the data are to be stored, antecedent rainfall data calculated by CARP for the required number of days previous to the event are written onto ORACLE table CARAIN3.

3.4.5.7 *Getting a print-out of the data loaded onto ORACLE tables*

Details of the results of the analysis of the rainfall are sent to file COUTP LISTING A, and lineprinter listings will arrive in your pigeon-hole automatically. All data entered on ORACLE tables are stored in file CLOAD LISTING A. Print-out detailing these data can be obtained by typing PRINT CLOAD LISTING A when command has returned to CMS. Lineprinter output will be sent to your pigeon-hole.

3.4.5.8 *CARP subroutines*

CARP uses several of the FORTRAN subroutines described in Section 4.4 to perform various database operations. These include

- AROFF - used to retrieve the catchment area from ORACLE table SWA.STATION_DETAIL.
- CAR3DEL - used to delete CARP catchment average rainfall for specified catchment, date/time and interval from ORACLE table CARAIN3.
- CAR3ON - used to load CARP catchment average rainfall onto ORACLE table CARAIN3.
- CODEL - used to delete all entries for a given event and data type from ORACLE table COMPONENT.
- COFF10 - used to retrieve the flow start date for a given event from ORACLE table COMPONENT.
- COFF11 - used to retrieve collected_by code for a given event from ORACLE table EVENT.
- COLLECT - used to retrieve collected_by code for a given code from ORACLE table COLLECTOR.

- COMPON - used to load an entry onto ORACLE table COMPONENT.
- CSTOFF - used to retrieve an ordered list of coastline coordinates from ORACLE table DIG.COAST_XY_GB_250K.
- GSOFF - used to retrieve easting and northing of gauging station from ORACLE table SWA.STATION_DETAIL.
- PSURCH - used to check that the catchment boundaries have been defined using subroutine POLYON and loaded onto ORACLE table SEARCH_POLY.
- QSURCH - used to check that the quadrilateral corners have been loaded onto ORACLE table CATCHMENT_QUAD.
- QUOFF - used to retrieve the corners of the catchment quadrilateral from ORACLE table CATCHMENT_QUAD.
- RGNAME - used to retrieve raingauge name from ORACLE table RAIN_GAUGE_DETAIL.
- RGSOFF - used to retrieve raingauge features from ORACLE table RAIN_GAUGE_SITE.
- RNGLIST - used to compile a list of raingauges for a given catchment from a search with a specified factor directly from ORACLE tables RAIN_GAUGE_SITE and SEARCH_POLY.
- RNIOFF - used to retrieve rainfall and data interval for a gauge given the start and end times from ORACLE table RAIN.
- RNOFF - used to retrieve rainfall for a gauge given the start and end times and data interval from ORACLE table RAIN.
- SALT - used to retrieve the raingauge SAAR and altitude from ORACLE table RAIN_GAUGE_SITE.
- SAROFF - used to retrieve the catchment AAR from ORACLE table SWA.CATCHMENT_CLIMATE.
- STNOFF - used to retrieve the river name and location corresponding to the catchment number from ORACLE table SWA.STATION_NAME.

3.5 *PERCH - Plotting Event Rainfall and Catchment Hydrograph*

3.5.1 Forward

Graphical representation of information is useful for many reasons, but essentially it provides a method of viewing the data and displaying important characteristics, which can often reveal errors or inconsistencies not apparent from columns of numbers eg. timing errors between rainfall and flow, discrepancies between hourly gauges, or the possible presence of snowmelt. Any one of these things may cause an event to be rejected.

3.5.2 Introduction

PERCH (Plotting Event Rainfall and Catchment Hydrograph) is a program which produces a graphical display of a lumped rainfall-runoff event on a particular catchment. River discharge is plotted against time to depict the catchment hydrograph, and hourly rainfalls through the event are plotted as hyetographs of both catchment average and point estimates. An example is given in Figure 3.5.1. The program can plot up to four point rainfall hyetographs, but in some cases rainfall values may not be available for any point locations, in which case only the flow hydrograph and catchment average rainfall are plotted. If either of these are not available the program will not run. A single subroutine simultaneously retrieves relevant flow and rainfall data in time series order from the database. This data can either be displayed on the screen, or sent to a plotter/printer for output on paper. The program does not produce any new statistics.

3.5.3 Quick guide

1. Log on and link to FLOOD's D-disk

2. Type PERCH

3. Wait for program to load

4. Input the required type of output device. The choice is between

   - the SCREEN,
   - the BENSON plotter,
   - the IBM 3812 LASER printer A4 size (3812A4),
   - the IBM 3812 LASER printer A6 size (3812A6) - used specifically for the production of an A6 size PERCH of the event on the catchment involving the greatest peak flow (see Section 3.3.5.5 for more details).

5. Input the 5-digit catchment number. Type a negative number to exit from the program. Type 0 to input information from a file, then input the file name, type and mode, and then go to step 9 (see Section 3.3.5.7 for file format details). If you make a mistake, there will be a chance to correct it later.

6. Input the date on which the flow data for the event starts in the standard ORACLE format DD-MON-YYYY.

7. Input the source of the catchment average rainfall data. This will be one of

   - AUTOS,
   - SCRAP,
   - CARP,
   - RADAR.

8. Input the 2-digit project code eg. 11, 12, 13 or other.

9. The screen displays the input details for checking. Type Y and press < ENTER > if all the entries are correct, and go to step 10. If changes are required type N, and press < ENTER >, and answer all the questions again.

10. If you are plotting on the screen, the screen will clear and the plot will appear after a few moments. Press < ENTER > to continue.

11. To exit from the program select a catchment number of -1. If the plots have been displayed on the screen `MORE..` will appear in the bottom right of the screen. Press < CLEAR > and then < ENTER > to terminate the program.

12. If you are sending your plots to the BENSON continue entering event details until finished, then exit from the program. Next type `BENPLOT` and press < ENTER > . Enter any special messages for the operators and press < ENTER > . The plots will eventually arrive in your pigeon-hole.

13. If you are sending your plots to the IBM 3812 LASER printer continue entering event details until finished, then exit from the program. Each event will be in your filelist as a separate file of the type 13812Pnn LISTAPA A where nn refers to a page number. Each file must be sent to the printer by the command `PRT3812 fn ft fm (ORIENT E`. Details of how to utilise the CMS command `FILELIST` to do this may be found in the February 1990 Wallingford User Note. The plots will soon arrive in your pigeon-hole.

14. Print-out detailing the data plotted can be obtained by typing `PRINT PERCH OUTP A`. Output will be sent to your pigeon-hole.

3.5.4 PERCH Pre-requisites

In order to produce a successful event plot, some data associated with the catchment for the event must have been loaded onto the relevant tables in the ORACLE database.

- RIVER NAME AND LOCATION - Each gauging station in the UK, which is recognised by the Surface Water Archive, is given a catchment number. This number is stored against the names of the station and the water course in ORACLE table SWA.STATION_NAME. The PERCH program cannot be run on catchments which are not recognised by the Surface Water Archive.
- AREA - The drainage area of the catchment is required because the flow values are expressed in millimetres over the catchment in addition to cubic metres per second. Values of the drainage area for all catchments recognised by the Surface Water Archive are stored on ORACLE table SWA.STATION_DETAIL.
- EVENT NUMBER AND COMPONENT ENTRIES - Each event is given a unique identification number. Details of the components which make up the event are stored on ORACLE table COMPONENT. These include catchment and raingauge numbers, and start and end times of point rainfalls, catchment average rainfalls and flows on ORACLE tables RAIN, CARAINn and FLOW respectively. An entry for the flow data must be made manually in the COMPONENT table, usually when the EVENT NO is being allocated. If the catchment average rainfall was derived using CARP, an entry will have been set up automatically in the COMPONENT table, otherwise an entry must be made manually. A fatal error results if either of these component entries are not found. Point rainfall data are not essential for running PERCH, and if none are found a message to this effect will be sent, and the words `NO HOURLY GAUGES AVAILABLE` will be printed in the appropriate place on the plot.

3.5.5 Running PERCH

PERCH is run by typing `PERCH` when linked to user-id FLOOD on the IBM.

```
THANKYOU FOR CHOOSING PERCH
 PLEASE WAIT WHILE LOADING
```

appears on the screen. It takes a few moments to link the compiled subroutines together. This is followed by

```
DEVICE TYPE  (A6)
  BENSON SCREEN 3812A4 or 3812A6
```

to which you should input the required type of output device in the manner shown.

3.5.5.1 Form filling

Input of information to PERCH is achieved by answering questions which appear on the screen, though control may be transferred to an input file if more convenient (see Section 3.3.5.7). The questions have the following form

```
            PPPP      EEEEE    RRRR      CCCC     H   H
            P   P     E        R   R    C         H   H
            P   P     E        R   R    C         H   H
            PPPP      EEE      RRR      C         HHHHH
            P         E        R  R     C         H   H
            P         E        R   R    C         H   H
            P         EEEEE    R    R    CCCC     H   H

          PLOTTING    EVENT   RAINFALL & CATCHMENT HYDROGRAPH

          CATCHMENT NO (I5) eg. 19001, -ve number to quit
                               input 0 for input from file
19001

          START DATE   (A11) eg. 13-AUG-1966
13-AUG-1966

          DATA SOUCE   (A5)
        either AUTOS SCRAP CARP or RADAR
CARP

           PERCH INPUT DATA ENTRY (contd)

          PROJECT CODE  (I2)
         11  FSR original event
         12  FSSR16 event
         13  Post FSSR16 event
         etc.
11

      YOU HAVE SELECTED DATA FOR CATCHMENT NUMBER 19001 FROM CARP
                    FOR THE PERIOD STARTING 13-AUG-1966
                  FOR PROJECT CODE 11 FSR original event
                        TO BE PLOTTED ON THE BENSON

                            ALL CORRECT?  (Y/N)
```

- DEVICE TYPE - Event plots can be displayed on the terminal by selecting SCREEN, sent to the BENSON plotter by typing BENSON, or sent to the IBM 3812 LASER printer by typing 3812A4. It is not possible to swap between devices during a single run, though when plotting on the screen the data are stored in a file which can be sent to the BENSON later; note however that it is not possible to plot a subset of the events displayed on the screen as the file contains data from the entire PERCH run. If you are sending plots to the IBM 3812 LASER printer there may be disk space restrictions as each event requires around 100 blocks, so make sure that you have sufficient room before you start. This may be achieved by sending large files not in current usage to your reader (by typing SENDF fn ft fm TO *) and erasing the originals from your filelist. There are various advantages and disadvantages of the two forms of hardcopy output: the BENSON style PERCH is in colour, but takes at least a day to appear and then requires trimming, whilst the IBM 3812 style PERCH is in black and white, but appears virtually instantaneously on A4 paper.

- CATCHMENT NO - This is the 5-digit Surface Water Archive reference number given to each gauging station in the UK. If the number is invalid INVALID CATCHMENT NO will appear, press <ENTER> to try another number or type STOP and press <ENTER> to quit. If the number is valid but no details of the river or station name are recorded STATION / RIVER NAME NOT FOUND will appear, press <ENTER> to try another number or type STOP and press <ENTER> to quit. The drainage area of the catchment is also required and if this value is not held on the database CATCHMENT AREA NOT FOUND will appear. The user is then prompted for the area, which should be entered in F6.1 format. Pressing <ENTER> without entering a value exits from the program.

- START DATE - This is the date of the first day of the flow data for the event. The date must be entered in the standard ORACLE format DD-MON-YYYY. The start date and project code (see below) are used to identify the particular event number.

- DATA SOURCE - This indicates the method used to derive the catchment average rainfall profile for the event. At present the data come from one of four sources, three of which are computerised methods of average point rainfalls: AUTOS (short for Autostorm, used in Flood Studies Report analysis), SCRAP and CARP. The fourth source, RADAR, indicates data from weather radar which may be available directly as a catchment average. The DATA SOURCE is used to identify the relevant EVENT_CARAIN view for catchment average rainfall data retrieval: CARAIN1 holds data from Autostorm, CARAIN2 data from SCRAP, CARAIN3 data from CARP, and CARAIN4 data from weather radar, so consequently there are four versions of the EVENT_CARAIN view (EVENT_CARAIN1, etc).

- PROJECT CODE - The same sequence of rainfall and flow values may be constituent parts of several different events. For example one event data application may require a long flow recession, whilst for another it is sufficient to truncate the flow after the peak. Clearly they must be assigned different event numbers as they use different amounts of data, but they cannot be uniquely identified by the start date of the flow. To overcome this ambiguity each event is also identified by a 2-digit PROJECT CODE. The present codes include 11 for FSR original events, 12 for FSSR16 events, 13 for Post FSSR16 events, and others. Full details are provided on ORACLE table COLLECTOR. Code 20 (complex events) may constitute two or more consecutive events of FSR or FSSR16 type.

3.5.5.2 *Plotting on the screen*

If you want to plot on the screen you must be logged on to a 3179 graphics terminal. Enter SCREEN when DEVICE TYPE is asked for. When all the input information is correct, press Y, and the screen will clear and the plot will appear after a few moments. Press <ENTER> to continue. Once the plots have been examined on the screen they can be sent to the BENSON plotter without re-running PERCH. Exit from the program by selecting a catchment number of -1, and when command has returned to CMS type BENPLOT and press <ENTER>. Input any special messages for the operators and press <ENTER>. The plotting information will be sent to the plotter, and the plots will eventually arrive in your pigeon-hole.

3.5.5.3 *Sending plots to the Benson plotter*

If you want to send your plots to the BENSON, without first displaying them on the screen, input BENSON when DEVICE TYPE is asked for. When all the input information is correct, press Y, and you will then be prompted for details of the next event. Continue entering event details until finished, then exit from the program by selecting a catchment number of -1. The information for the plotter will be held in file B1645 PLOTCODE A. When command has returned to CMS type BENPLOT and press <ENTER>. Input any special messages for the operators and press <ENTER>. The plotting information will be sent to the plotter, and the plots will eventually arrive in your pigeon-hole.

3.5.5.4 *Sending plots to the IBM 3812 Laser printer (A4 size)*

If you want to send your plots to the IBM 3812, input 3812A4 when DEVICE TYPE is asked for. When all the input information is correct, press Y. On the screen will appear the words

```
Page size is A4 Landscape
IBM 3812 page printer - page nn
```

where nn refers to a page number ie. your first event will be page number 1, and so on. You will then be prompted for details of the next event. Continue entering event details until finished, then exit from the program by selecting a catchment number of -1. The information for the printer will be held in a separate file for each event, of the type I3812Pnn LISTAPA A where nn refers to the page number. When command has returned to CMS type PRT3812 fn ft fm (ORIENT E for each file. Details of how to utilise the CMS command FILELIST to do this may be found in the February 1990 Wallingford User Note. The files will be sent to the printer, and the plots will soon arrive in your pigeon-hole.

3.5.5.5 *Producing Representative Basin Catalogue style plots (A6 size)*

The Representative Basin Catalogue includes for each catchment an A6 size PERCH of an example event. This event is chosen as the one on the catchment of project code 11, 12 or 13 involving the greatest peak flow. This event may be plotted from the EXEC CATALOG where it is included in the catchment's Representative Basin Catalogue entry, or may be plotted alone by selecting DEVICE TYPE 3812A6, and entering only the required catchment number. If a suitable event is not found, the program will terminate, otherwise on the screen will appear the words

```
     Page size is A6 Landscape
     IBM 3812 page printer - page  1
```

The program will then terminate. The information for the printer will be held in file I3812P01 PSEG3820 A. In order to be printed this file must be included in a SCRIPT/VS document by using the .si SCRIPT control word. The file PERCH SCRIPT D contains the single line

```
     .si I3812P01
```

In CMS type `PRT3812 PERCH SCRIPT A (SCRIPT XMIT`. The file will be sent to the printer, and the plot will soon arrive in your pigeon-hole.

3.5.5.6 Getting a print-out of the data used in the plotting

All data used to derive the plots are stored in file PERCH OUTP A. Print-out detailing these data can be obtained by typing `PRINT PERCH OUTP A` when command has returned to CMS. Lineprinter output will be sent to your pigeon-hole.

3.5.5.7 Entering input information from a file

To avoid the tedious undertaking of entering a large number of events using the input form, the relevant information can be entered from a file. A separate file must be set up for each catchment. Entry of information from file is initialised by entering a CATCHMENT NO of 0, and then the file name, type and mode. The file must have the following format

```
1st line        I10                 Catchment number
2nd line        A5                  Blank line, or five spaces
3rd line        I10                 Source of data: 1 for AUTOSTORM, 2 for
                                    SCRAP, 3 for CARP, 4 for RADAR
4th line        A11,I4              Date at start of event, project code:
                                    11 for FSR original event, 12 for
                                    FSSR16 event, 13 for post-FSSR16 event,
                                    etc
5th and subsequent lines as line 4
```

For example

```
     19001

         1
12-AUG-1978  13
```

If the project code is left blank, 13 is assumed.

3.5.6 PERCH subroutines

PERCH uses several of the FORTRAN subroutines described in Section 4.4 to perform various database operations. These include

- AROFF - used to retrieve the catchment area from ORACLE table SWA.STATION_DETAIL.
- COFFX - used with DEVICE TYPE 3812A6 to find the number of the event on the catchment of project code 11, 12 or 13 involving the greatest peak flow.
- COFF1 - used with all DEVICE TYPEs except 3812A6 to find the event number corresponding to the start date and project code from ORACLE table COMPONENT.
- COLLECT - used to retrieve collected_by code for a given code from ORACLE table COLLECTOR.
- FROFF1 - used to retrieve flow data from ORACLE view EVENT_FLOW and catchment average rainfall data from the relevant EVENT_CARAINn view identified from the DATA SOURCE input.
- QUOFF - used to retrieve the centroid of the catchment quadrilateral from ORACLE table CATCHMENT_QUAD.

- RROFF - used to retrieve point rainfall data from ORACLE view EVENT_RAIN which holds point rainfall data from table RAIN for the raingauge for the period specified in the COMPONENT table for the event.
- STNOFF - used to retrieve the river name and location corresponding to the catchment number from ORACLE table SWA.STATION_NAME.

3.6 RUNLPROG FSR Unit Hydrograph Programs

Exec RUNLPROG runs the flow and rainfall separation program LP5 and the unit hydrograph derivation program LP6. The programs are called LPROGS because they were written by Mike Lowing for the flood study project. The exec expects a file name (name, type & mode) but will prompt for it if it is not entered as an argument. The file identifies a catchment and specifies the start date of the event. The time-series data are stored on ORACLE tables and are accessed directly by the program; the necessary data and component entries must have been entered previously. The date given in the file refers to the start date of the flow data; it is assumed that only one event will have the same catchment and flow gauge number (if this ever turns out not to be the case the resulting ambiguity will need to be resolved). The catchment area must be loaded on table SWA.STATION_DETAIL. The various indicators in the data file ensure the correct event and rainfall data are selected.

The format of the file is

```
1st line        I10             Catchment number
2nd line        2I10            Dummy, source of data: 1 for AUTOSTORM,
                                2 for SCRAP, 3 for CARP, 4 for RADAR
3rd line        I10             Source of data: 1 for AUTOSTORM, 2 for
                                SCRAP, 3 for CARP, 4 for RADAR
4th line        A11,I4          Date at start of event, project code:
                                11 for FSR original event, 12 for
                                FSSR16 event, 13 for post-FSSR16 event,
                                etc
5th and subsequent lines as line 4
```

For example

```
     19001
                   1
         1
12-AUG-1978  13
```

The programs run separately on the whole batch of events, the separation program runs first (obviously). A data file transfers separated data between the programs. The unit hydrograph derivation program requests a method code to be entered, 2 for matrix inversion (with or without smoothing) or 1 for harmonic analysis.

Data channels set up are

```
Lprog5
     FILEDEF 41 PRINTER
     FILEDEF 48 DISK       LP6 INPUT A (deleted after execution)
     FILEDEF 46 DISK       LP5 SUMMARY A
     FILEDEF 40 PRINTER
     FILEDEF 7 DISK        User's input
     FILEDEF 6 TERM

Lprog6
     FILEDEF 39 DISK       UH OUT39 A'
     FILEDEF 42 PRINTER
     FILEDEF 44 DISK       UH OUT44 A'
```

3.6.1.1 EVCOPY (terminal)

EVCOPY creates a new event identical to a given event except for application and collector codes. Selected components can also be copied across to the new event. The routine sets up an event cross-reference on ORACLE table EVENT_REFERENCE between the old and new event. This SQL*plus routine is particularly useful when dealing with long, complex events during analysis. The complete event can be copied to a new event number under collector code 20 for "rainfall-runoff modelling", whilst the original event may be truncated for FSR-type analysis purposes. The job can be run more than once to create several short events from one long one; when doing this use the original long event as the old event each time.

To use, log on and link to FLOOD's D-disk, log on to ORACLE, and type

```
START EVCOPY.SQL
```

and enter the old event number, new application, new collector code, and the data types and intervals of the components to be copied across to the new event when prompted. The new event number will be printed out at the end of the routine. eg.

```
START EVCOPY.SQL
Enter value for new_application:
1
Enter value for new_collected_by:
20

1 record created.

Enter value for old_event:
4296

1 record updated.

1 record created.

Enter value for data_types_to_be_copied:
1,4,10,13,17
Enter value for data_intervals_to_be_copied:
1,24

12 records created.

NEW_EVENT_NUMBER
----------------
            4359
```

Note that it will not be possible to adjust the rainfall of the truncated event using CARP as the program will now find overlapping events. Component dates must be adjusted as described in EVDEFS2 (Section 4.3.2).

3.7 GUPPY - Graphs of Unit hydrograph Parameters and Percentage Yield

3.7.1 Forward

The Flood Studies Report rainfall-runoff model is used throughout the UK to estimate design floods when the entire flood hydrograph is required. The rainfall-runoff transformation model essentially has two parameters: the percentage runoff, PR, which controls the proportion of rainfall contributing to the volume of the flood hydrograph, andt the time to peak of the unit hydrograph, Tp, which controls the shape of the design hydrograph. When a number of historical flood events on a catchment have been analysed it is useful to examine the variation in these parameters and the events characteristics.

3.7.2 Introduction

GUPPY (Graphs of Unit hydrograph Parameters and Percentage Yield) comprises of a set of programs implemented on the IBM computer at Wallingford, under user-id FLOOD. GUPPY produces a graphical display of the variation in Flood Studies Report unit hydrograph parameters time-to-peak and percentage runoff across all events analysed for the particular catchment selected. The display consists of two sets of graphs, with three individual graphs in each set. The first set of graphs display percentage runoff for each event plotted against initial flow, a catchment wetness index and the total event rainfall. Percentage runoff is not a function of a single variable, and a broad indication of the its relationship with total rainfall and catchment wetness used in the model is given by curves in the second and third graphs. An example is given in Figure (X.Y.Z). In the second set unit hydrograph time-to-peak for each event is plotted against unit hydrograph width, event peak flow and unit hydrograph peak flow separately on the three graphs (see Figure X.Y.Z). Curves are drawn on the first and third graphs depicting the relationship between the unit hydrograph variables assumed in the rainfall-runoff model. Data are retrieved from the Institute of Hydrology flood event archive of analysed flood events. This contains events analysed for the Flood Studies Report, Flood Studies Supplementary Report 16, and subsequent studies such as the examination of percentage runoff on Scottish catchments (Boorman et al, 1988). Details of the events analysed for FSSR16 are given in IH Report 94 (Boorman, 1985). The program displays the graphs either on the screen of a graphics terminal or they are sent to a plotter for printing on paper. The program does not produce any new statistics.

All the data required for running GUPPY are stored on tables in the ORACLE database. The majority of these are under the user-id FLOOD. Some additional tables, identified by the prefix SWA, eg. SWA.STATION_DETAILS, belong to the Surface Water Archive, and these contain the most up-to-date information regarding gauge sites which comes directly from regional NRAs.

3.7.3 Quick guide

1. Log on, or link, to Flood

2. Type GUPPY

3. Wait for program to load

4. Enter the catchment number, which must consist of five figures, and press <ENTER>. Enter a negative number if you wish to exit from the program.

5. The type of plotting device must be selected next. The choice is between the BENSON plotter and the SCREEN.

6. If you are plotting on the SCREEN you must be logged on to an IBM 3179 terminal. Wait for the screen to clear and for the cursor to reach the top right-hand corner of the screen, then press <ENTER>. The plot will appear after a few moments. Press <ENTER> to continue. Second and subsequent plots appear automatically, without the need to press <ENTER>.

7. To exit from the program select a catchment number of -1. If the plots have been displayed on the screen MORE.. will appear in the bottom right of the screen. Press <CLEAR> and then <ENTER> to terminate the program.

8. If you are sending your plots to the BENSON continue entering event details until finished, then exit from the program. Next type BENPLOT. Enter any special messages for the operators and press < ENTER >. The plots will eventually arrive in your pigeon hole.

9. Print-out detailing the data plotted can be obtained by typing PRINT GUPPY OUTP A. Output will be sent to your pigeon-hole.

3.7.4 GUPPY Pre-requisites

In order to produce a successful event plot, some data associated with the catchment for the storm event must have been loaded onto the relevant databases on the IBM.

3.7.4.1 River name and location

Each gauging station in the UK, which is recognised by the Surface Water Archive, is given a reference number. This number is stored against the names of the station and the water course in ORACLE table SWA.STATION_NAME. Catchment data cannot by stored for catchments which are not recognised by the Surface Water Archive. A fatal error results if the input reference number is not found in the table. See Section X.Y.Z.1 for details.

3.7.4.2 Analysed events

At least one event must have been analysed by the Flood Studies Report unit hydrograph method. The results of the analysed events must be stored on ORACLE table POST_FSSR16_RESULT. Table FSR_RESULT and FSSR16_RESULT contain the results of analysis for the the original Flood Studies Report and Flood Studies Supplementary Report 16 respectively. Only data from FSSR16_RESULT and POST_FSSR16_RESULT are used in the plots. A fatal error results if no event data are found in either of these two tables.

3.7.5 Running GUPPY

GUPPY is run by typing GUPPY when logged on to, or linked to, the user-id FLOOD on the IBM.

```
THANKYOU FOR CHOOSING GUPPY
 PLEASE WAIT WHILE LOADING
```

appears on the screen. It takes a few moments to link the compiled subroutines together.

3.7.5.1 Form filling

The input of the required information to GUPPY is achieved by answering the questions which appears on the screen. Control may be transferred to an input file if several plots are required. The questions have the following form

```
      GGGGG       UU    UU      PPPPPP        PPPPPP     YY    YY
     GG   GG      UU    UU      PP   PP       PP   PP    YY    YY
     GG           UU    UU      PP   PP       PP   PP     YY  YY
     GG   GGG     UU    UU      PPPPPP        PPPPPP       YYYY
     GG   GG      UU    UU      PP            PP            YY
     GG   GG      UU    UU      PP            PP            YY
      GGGGG        UUUUUU       PP            PP            YY

      Graphs of Unit hydrograph Parameters and Percentage Yield

<ENTER> CATCHMENT NUMBER (I5) eg. 19001   -ve number to quit

<ENTER> DEVICE TYPE either SCREEN or BENSON
```

CATCHMENT No This is the Surface Water Archive reference number given to each gauging station in the UK. Enter the catchment number in I5 format (five digits) and press return. If the number is invalid INVALID CATCHMENT No will appear, press < ENTER > to try another number or type STOP and press < ENTER > to quit. If the number is valid but no details of the river or station name are recorded STATION / RIVER NAME NOT FOUND will appear, press <ENTER> to try another number or type STOP and press <ENTER> to quit.

DEVICE TYPE The GUPPY plots can either be displayed on a terminal by selecting SCREEN or sent to the BENSON plotter to obtain a hard copy by typing BENSON. It is not possible to swap between devices during a single run however, even when plotting on the screen the plotting data are stored in a file called B1645 PLOTCODE A which can be sent to the BENSON plotter to obtain a hard copy. Note that it is not possible to plot a subset of the events displayed on the screen, the file contains data from the entire GUPPY run.

3.7.5.2 *Plotting on the screen*

If you require a plot on a computer terminal you must be logged on to a 3179 graphics terminal. Input SCREEN when DEVICE TYPE is asked for. Wait for the screen to clear and for the cursor to reach the top right-hand corner of the screen, then press <ENTER>. The plot will appear after a few moments. Press <ENTER> to continue to the next event. Second and subsequent plots appear automatically, without the need to press <ENTER>. To exit from the program select a catchment number of -1. PROGRAM TERMINATING PLEASE WAIT .. will appear on the screen, followed by MORE.. in the bottom right of the screen. Press <CLEAR> and the <ENTER> to complete termination of the program and return to CMS.

Once the plots have been examined on the screen they can be plotted on the BENSON plotter without re-running GUPPY. When command has return to CMS type BENPLOT. Enter any special messages for the operators and press <ENTER>. The plotting information will be sent to the plotter and the plot will eventually find its way to your pigeon-hole.

3.7.5.3 *Sending plots to the Benson plotter*

If you want to send your plots to the BENSON, without displaying them on the screen, input BENSON, when DEVICE TYPE is asked for, then type Y if all the input information is correct, and press <ENTER>. You will be prompted for details of the next event. Continue entering events details until finished, then exit from the program by selecting a catchment number of -1. The information for the plotter will be held in file B1645 PLOTCODE A. PROGRAM TERMINATING PLEASE WAIT .. will appear on the screen, followed by MORE.. in the bottom right of the screen. Press <CLEAR> to complete termiantion of the program and return to CMS. When command has return to CMS type BENPLOT. Enter any special messages for the operators and press <ENTER>. The plotting information will be sent to the plotter and the plot will eventually find its way to your pigeon-hole.

3.7.5.4 *Getting a print-out of the data used in the plotting*

All data used to derive the plots are stored in a file called GUPPY OUTP A. A hard-copy of these data can be obtained by typing PRINT GUPPY OUTP A after exiting from GUPPY. Output will be sent to your pigeon-hole.

3.7.6 How GUPPY works

3.7.6.1 *Station name*

GUPPY uses subroutine STNOFF to identify the name of the gauging station and the watercourse from the catchment number. If the catchment number given is not recognised by the Surface Water Archive an error message is printed and the option is available to enter another catchment number or exit from the program. For details see Section X.Y.Z.1.

3.7.6.2 *Flood event analysis results*

GUPPY retrieves the results of unit hydrograph analysis of events on the catchment using subroutine UHRES. This subroutine returns the following details for each analysed event found in either view UH_ANALYSIS_RESULT which is a union of tables FSSR16_RESULT and POST_FSSR16_RESULT

(1) Date of event

(2) Peak discharge
(3) Percentage runoff
(4) Total event rainfall
(5) Soil moisture deficit at start of event
(6) Antecedent precipitation index
(7) Flow at start of event
(8) Soil moisture deficit at start of event
(9) Time to peak of unit hydrograph
(10) Peak flow of unit hydrograph
(11) Width of unit hydrograph at the peak flow

From these characteristics all important unit hydrograph rainfall-runoff model parameters can be calculated, such as the standard percentage runoff, SPR.

3.7.6.3 Fraction of catchment urbanised

In the Flood Studies Report unit hydrograph method percentage runoff, PR, is function of three elements, the standard percentage runoff SPR which depends on soil type, dynamic percentage runoff DPR which depends on the storm characteristics, and URBAN, the fraction of the catchment under urban development. PR and the storm characteristics P and CWI, needed to calculate DPR, are stored in the result files, so only URBAN is needed to calculate SPR. URBAN is stored on ORACLE table SWA.CATCHMENT_LANDFORM and can be retrieved using subroutine URBOFF.

3.7.6.4 Plotting the graphs

GUPPY produces two sets of graphs, each with three individuals.

Graphs of variation in percentage runoff: The first set of graphs display percentage runoff for each event plotted against (1) initial flow, (2) a catchment wetness index, CWI, and (3) the total event rainfall, P. In the Flood Studies Report model percentage runoff is a function of both total rainfall and catchment wetness and a single value of SPR. Therefore it is not possible to show the exact relationship used in the model on a graph of PR against a single variable. However, if one variable can be assumed to remain constant the consequence of changing the other variable can be depicted. In graph (2) a blue curve shows how PR increases with increasing CWI if the total rainfall is always less than, or equal to, 40mm. When this is so, P is only a function of CWI and SPR, having the form

$$PR = SPR + 0.25\,(CWI - 125)$$

Thus at CWI = 125, PR = SPR.

In graph (3) a blue curve shows the rate of increase in PR as P increases assuming that CWI is constant at 125. When this is so, PR is only a function of P and SPR, having the form

$$PR = SPR + 0.45\,(P - 40) ** 0.7 \quad \text{for } P > 40$$

Note that the curve is horizontal to P = 40mm, since up to this value of P, PR = SPR.

Graphs of variation in time-to-peak: In the second set of graphs unit hydrograph time-to-peak, Tp, for each event is plotted against three variables: (1) unit hydrograph width, W, (2) event peak flow and (3) unit hydrograph peak flow, Qp. In the Flood Studies Report model both W and Qp are functions Tp. Curves are drawn on the first and third graphs depicting these relationships. In graph (1) a blue curve shows how W increases with increasing Tp. The relationship is given by the function

$$W = (-0.00834\,Tp + 1.399)\,Tp$$

Note that when using a simple triangular unit hydrograph this equation reduces to W = 1.26 Tp.

In graph (3) a blue curve shows the rate of decrease in Qp as Tp increases. The function relating Qp and Tp is

$$Qp = (2.587\,Tp + 162.2)\,/\,Tp$$

Note that when using the design procedure the function is simplified to Qp = 220 / Tp.

3.8 *IEM4 - The Isolated Event Model Version 4*

3.8.1 Forward

The Flood Studies Report recommends the use of the unit hydrograph method of flood estimation when a design flood hydrograph is required. For other applications, such as the flood warning system on the River Tyne at Haddington, a conceptual model of the rainfall-runoff relationship has been used. The fundamental difference between these two models is that in the unit hydrograph model this relationship is linear whereas in the IEM it is non-linear. At present application of the IEM is confined to gauged catchments.

3.8.2 Introduction

IEM4, the Isolated Event Model version 4, comprises of a set of programs implemented on the IBM computer at Wallingford, under user-id FLOOD. IEM4 produces a graphical display of a hydrological event in terms of its rainfall and runoff on a particular catchment. Both the observed river discharge and that predicted by the IEM, given the observed rainfall input, are plotted against time to depict the catchment hydrograph. Catchment average hourly rainfalls through the event are plotted as a hyetograph. An example is given in Figure 1.1.1. For successful application of the model the observed flow hydrograph and catchment average rainfall must be available, otherwise the program will not run. A large database is held at the Institute of Hydrology containing event data for many catchments which include flow values, and catchment average rainfalls derived, in some cases by CARP (see Section X). Details of these various components which make up each event are also stored, thus when an event is requested the relevant data can be easily assembled by the computer. The program then displays these data either on a screen or they are sent to a plotter for printing on paper. The program does not produce any new statistics.

All the data required for running IEM4 are stored on tables in the ORACLE database. The majority of these are under the user-id FLOOD. Some additional tables, identified by the prefix SWA, eg. SWA.STATION_DETAIL, belong to the Surface Water Archive, and these contain the most up-to-date information regarding gauge sites which comes directly from regional NRAs.

3.8.3 Quick guide

1. Log on, or link, to Flood

2. Type IEM4

3. Wait for program to load

4. Enter the catchment number, which must consist of five figures, and press <ENTER>. Enter a negative number if you wish to exit from the program. The input information can be entered into a file. To select this form of data entry type 0 and press <ENTER>, then goto step 9.

5. Enter the date on which the flow data for the event starts in the format 13-AUG-1966.

6. Now input the source of the catchment average rainfall data. This will be either AUTOS, SCRAP, CARP or RADAR.

7. Next input the 2-digit project code, either 11, 12, 13 or 20.

8. The type of plotting device must be selected next. The choice is between the BENSON plotter and the SCREEN. Goto to step 10.

9. If a catchment number zero was entered in step 4, next input the name of the file containing the input instructions (see section XXXXXX for format details).

10. The screen displays the input details for checking. Type Y or just press <ENTER> again if all the entries are correct. If changes are required input N, press <ENTER> and answer all of the questions again.

11. If current parameter values are available these will be displayed for checking; if not input is requested. You will also need to decide whether these parameters are to be optimised. Type Y,

or press <ENTER>, if all the entries are correct. If changes are required input N, press <ENTER> and input the values again. To select the next set from the input file type F and and press <ENTER>.

12. If you are plotting on the SCREEN you must be logged on to an IBM 3179 terminal. Wait for the screen to clear and for the cursor to reach the top right-hand corner of the screen, then press <ENTER>. The plot will appear after a few moments. Press <ENTER> to continue. Second and subsequent plots appear automatically, without the need to press <ENTER>.

13. To exit from the program select a catchment number of -1. If the plots have been displayed on the screen MORE.. will appear in the bottom right of the screen. Press <CLEAR> and then <ENTER> to terminate the program.

14. If you are sending your plots to the BENSON continue entering event details until finished, then exit from the program. Next type BENPLOT. Enter any special messages for the operators and press <ENTER>. The plots will eventually arrive in your pigeon hole.

15. Print-out detailing the data plotted can be obtained by typing PRINT IEM4 OUTP A. Output will be sent to your pigeon-hole.

3.8.4 IEM4 Pre-requisites

In order to successfully apply IEM4, some data associated with the catchment for the storm event must have been loaded onto the relevant databases on the IBM.

3.8.4.1 *River name and location*

Each gauging station in the UK, which is recognised by the Surface Water Archive, is given a reference number. This number is stored against the names of the station and the water course in ORACLE table SWA.STATION_NAME. Catchment data cannot by stored for catchments which are not recognised by the Surface Water Archive. A fatal error results if the input reference number is not found in the table. See Section X.Y.Z.1 for details.

3.8.4.2 *Area*

The drainge area of the catchment is required by IEM4 because the flow values are expressed in millimetres over the catchment in addition to cubic metres per second. Values of the drainage area for all catchments recognised by the Surface Water Archive are stored on ORACLE table SWA.STATION_DETAIL. If the program does not find the area, the user is prompted to enter it. Further details are given in Section X.Y.Z.2.

3.8.4.3 *Flow data*

Each event must have an associated sequence of flow data stored on ORACLE table FLOW. Each data item is identified by the catchment number, A fatal error results if no flow data are found.

3.8.4.4 *Catchment average rainfall data*

Each event must have an associated catchment average rainfall sequence identified by the catchment number and stored on one of the four ORACLE CARAIN tables. This may be derived by running CARP (see chapter X.) or, for example, may have come directly from a weather radar. A fatal error results if no catchment average rainfall data are found.

3.8.4.5 *Event number and component entries*

Each event is given a unique identification number. Details of the components which make up the event are stored on ORACLE table COMPONENT. These include catchment and raingauge numbers and start and end times of rainfall and flow data. An entry for the flow data and catchment average rainfall data must be made in the table and referenced by the event number. If the catchment average rainfall was derived using CARP, component entries will have been set up automatically. A component entry for the flow needs to be entered separately.

3.8.4.6 *Soil moisture deficit*

The antecedent catchment wetness can be indexed by the soil moisture deficit immediately prior to the storm. Catchment average soil mositure data are stored on ORACLE table CASMD. The entry relevant to each storm is referenced by an entry in the COMPONENT table.

3.8.5 Running IEM4

IEM4 is run by typing IEM4 when logged on to, or linked to, the user-id FLOOD on the IBM.

```
THANKYOU FOR CHOOSING THE
 ISOLATED EVENT MODEL V4
PLEASE WAIT WHILE LOADING
```

appears on the screen. It takes a few moments to link the compiled subroutines together.

3.8.5.1 *Form filling*

The input of the required information to IEM4 is achieved by answering the questions which appears on the screen. Control may be transferred to an input file if several plots are required. The questions have the following form

```
                 INSTITUTE OF HYDROLOGY

      IIIIIIII    EEEEEEEE   MMM    MMM         444
         II       EE         MMMM MMMM         4444
         II       EE         MM MMM MM        44 44
         II       EEEEEE     MM  M  MM       44  44
         II       EE         MM     MM     444444444
         II       EE         MM     MM           44
      IIIIIIII    EEEEEEEE   MM     MM           44

      ISOLATED      EVENT       MODEL    VERSION 4

      CATCHMENT No (I5) eg. 19001, -ve number to quit
                         input 0 for input from file

      START DATE   (A11) eg. 13-AUG-1966

      DATA SOUCE   (A5)
      either AUTOS SCRAP CARP or RADAR

       IEM4 DATA ENTRY (contd)

      PROJECT CODE  (I2)
     11  FSR original event
     12  FSSR16 event
     13  Post FSSR16 event
  or 20  Complex event

      DEVICE TYPE  (A6)
    either BENSON or SCREEN

     YOU HAVE SELECTED DATA FOR CATCHMENT NUMBER 19001 FROM AUTOS
                  FOR THE PERIOD STARTING 13-AUG-1966
   WHICH IS A TYPE 11 FSR original event TO BE PLOTTED ON THE SCREEN

                          ALL CORRECT?  (Y/N)
```

CATCHMENT NO This is the Surface Water Archive reference number given to each gauging station in the UK. Enter the catchment number in I5 format (five digits) and press return. If the number is invalid INVALID CATCHMENT NO will appear; press <ENTER> to try another number or type STOP and press <ENTER> to quit. If the number is valid but no details of the river or station name are recorded STATION / RIVER NAME NOT FOUND will appear, press <ENTER> to try another number or type STOP and press <ENTER> to quit. The drainage area of the catchment is also required to convert runoff to mm depth, if this value is not held on the archive CATCHMENT AREA NOT FOUND is displayed. The user is then prompted for the area, which should be entered in F6.1 format. Pressing return without entering a value exits from the program.

START DATE This is the date of the first day of the river flow data. The date must be specified in the standard ORACLE format DD-MON-YYYY, eg. 21-AUG-1982. The start date is used to identify the particular event.

DATA SOURCE This indicates the method used to derive the required catchment average rainfall profile. The data held at present come from one of four sources, three of these are computerised methods of average point rainfalls ie. AUTOS (short for Autostorm, used in analysis for the Flood Studies Report), SCRAP and CARP. The forth source, RADAR, indicates data from weather radar which may be available directly as a catchment average.

PROJECT CODE The same sequence of rainfall and flow values may be constituent parts of several different events. For example one event data application may require a long flow recession whilst for another it is sufficient to truncate the flow after the peak. Clearly they must be assigned by different event numbers as they use different amounts of data. However, they cannot be uniquely identified by the start date of the flow. To overcome this ambiguity each event is also identified by a PROJECT CODE. The present codes are 11 for FSR original events, 12 for FSSR16 events, 13 for Post-FSSR16 events and 20 for complex events. The latter type may constitute two or more consecutive events of FSR or FSSR16 type.

DEVICE TYPE The event plot can either be displayed on a terminal by selecting SCREEN or sent to the BENSON plotter to obtain a hard copy by typing BENSON. It is not possible to swap between devices during a single run however, even when plotting on the screen the plotting data are stored in a file called B1645 PLOTCODE A which can be sent to the BENSON plotter to obtain a hard copy. Note that it is not possible to plot a subset of the events displayed on the screen, the file contains data from the entire IEM4 run.

If current parameter values are available (if this is a second or subsequent run) these are displayed.

```
        CURRENT PARAMETER VALUES
        ************************

         DEL     7.31
         AC     24.30
         PERC
         PERI

      NO OPTIMISATION OF PARAMETERS REQUIRED

                ALL CORRECT?  (Y/N)

               IEM4 PARAMETER ENTRY
```

3.8.5.2 Plotting on the screen

If you require a plot on a computer terminal you must be logged on to a 3179 graphics terminal. Input SCREEN when DEVICE TYPE is asked for then type Y, if all the input information is correct, and press < ENTER >. Wait for the screen to clear and for the cursor to reach the top right-hand corner of the screen, then press < ENTER >. The plot will appear after a few moments. Press < ENTER > to continue to the next event or to quit. Once the plots have been examined on the screen the can be plotted on the BENSON plotter without re-running IEM4. When command has return to CMS type BENPLOT. Enter any special messages for the operators and press < ENTER >. The plotting information will be sent to the plotter and the plot will eventually find its way to your pigeon-hole.

3.8.5.3 Sending plots to the Benson plotter

If you want to send your plots to the BENSON, without displaying them on the screen, input BENSON when DEVICE TYPE is asked for, then type Y if all the input information is correct, and press < ENTER >. You will be prompted for details of the next event. Continue entering events details until finished, then exit from the program by selecting a catchment number of -1. The information for the plotter will be held in file B1645 PLOTCODE A. When command has return to CMS type BENPLOT. Enter any special messages for the operators and press < ENTER >. The plotting information will be sent to the plotter and the plot will eventually find its way to your pigeon-hole.

3.8.5.4 Getting a print-out of the data used in the plotting

All data used to derive the plots are stored in a file called IEM4 OUTP. A hard-copy of these data can be obtained by typing PRINT IEM4 OUTP A after exiting from IEM4. Output will be sent to your pigeon-hole.

3.8.5.5 Entering input information from a file

To avoid the tedious undertaking of entering a large number of events using the input form, the relevant information can be entered into a file. Entry of data from file is initialised by entering a CATCHMENT No of zero. The file must have the following format

```
03003 24-MAR-1984 CARP  13 SCREEN
  2.511  15.452   0.550   0.005 1
23006 13-DEC-1963 AUTOS 11
  1.923  21.321   0.620   0.004 1
```

The first line gives the station number, start date, source, project code and device type, whilst the second line gives the parameter values DEL, AC, PERC and PERI and finally the option of whether to optimise the parameters, (1) indicates yes, (2) no. The pair of lines must be repeated for each event. Subsequent lines are similar in format to these with the exception of the device type, which is only read from the first line.

3.8.6 How IEM4 works

3.8.6.1 Station name

IEM4 uses subroutine STNOFF to identify the name of the gauging station and the watercourse from the catchment number. If the catchment number given is not recognised by the Surface Water Archive an error message is printed and the option is available to enter another catchment number or exit from the program.

3.8.6.2 Area

IEM4 retrieves the catchment drainage area from ORACLE table SWA.STATION_DETAIL using AROFF. If the area is not found, the user is prompted to enter it.

3.8.6.3 Date

The date required by the program is the day of the beginning of the flow record for the event. If these flow data contain a long sequence prior to the start of the event, the start date may be the day before the actual event itself. The format DD-MON-YYYY is a standard ORACLE date format. This date is used to find the event number and hence the associated event data.

3.8.6.4 Event number

IEM4 calls subroutine COFF1 to find the event number. COFF1 searches though the COMPONENT table to find an entry for flow data beginning on the day specified by date. The event number associated with this entry is used provided that its corresponding entry in table EVENT also has the appropriate project code. An error results if no flow component entry is found.

3.8.6.5 Flow and catchment average rainfall data

IEM4 calls subroutine FROFF1. This retrieves data from view EVENT_FLOW and the relevant EVENT_CARAIN view. EVENT_FLOW contains flow data from table FLOW for the flow gauge identified by the catchment number for the period specified by the start and end dates given in the COMPONENT table entry for the event. At present there are four CARAIN tables (CARAIN1,CARAIN2 etc) which hold output from different catchment average rainfall estimation methods. Data from Autostorm (used for the Flood Studies Report) are stored on CARAIN1, output from SCRAP on CARAIN2 and on CARAIN3 from CARP. CARAIN4 holds data from weather radar which may be available directly as a catchment average. Consequently there are four versions of the

EVENT_CARAIN view. The data SOURCE input is used to identify the appropriate view. EVENT_CARAIN3 contains catchment average rainfall data from CARP for the period with the start and end dates specified by the relevant entry in the COMPONENT table for the event. If there are no data for the event, FROFF1 returns an error and the program terminates.

3.8.6.6 Point rainfall data

IEM4 uses subroutine RROFF to ascertain whether any point rainfall data are available for the event. If so data are retrieved from view EVENT_RAIN which contains point rainfall data from table RAIN for the period specified by the start and end dates given in the COMPONENT table entry for the event. If no gauges are found the message NO POINT RAINFALL DATA AVAILABLE FOR EVENT 1234 appears, where 1234 is the event number.

3.8.6.7 Parameter optimisation

If the option is selected, optimum values of the parameters entered will be calculated. The parameters are optimum in the sense that they provide the set which reproduces the observed hydrograph most closely. The closeness of fit is measured either just on the rising limb of the hydrograph, or across all ordinates, depending on whether option 1 or 2 is chosen. IEM4 uses the Nag optimisation routine E04JAF. This is a quasi-Newton algorithm for finding the minimum of a function subject to fixed upper and lower bounds. The function minimised is the sum of squared differences between observed and estimated flows either
Option 1 - on the rising limb of the hydrograph,
or
Option 2 - over all flow ordinates.

3.9 CATALOG (formerly BIGJOB)

The EXEC CATALOG produces the representative basin catalogue entry for a specified catchment. The command structure is

```
CATALOG Userid/password Catchment Format Version Copyright
```

Userid/password is a valid ORACLE user-id and password with permissions to the tables containing the data.
Catchment is the SWA catchment number
Format is the output format. Valid options are: NONE (for no output) TERM (for terminal), LINE (for lineprinter) and NICE (for the 3812 laser-type printer).
Version only applies when NICE format is selected, it can be LONG or SHORT. LONG includes all plots and SWA data descriptions that have to be prepared by FORTRAN programs, SHORT only accesses data directly through SQL*plus. LONG can take many (up to an hour) minutes to run.
Copyright adds an acknowledgements and copyright section to the end of the entry if required; this is done by putting COPY, and should be included in individual representative basin descriptions. If this section is not needed eg. for the catalogue itself, write NONE.

3.10 CADRE

CADRE (Catchment Average Daily Rainfall Estimation) generates a sequence of catchment average daily rainfalls, using Jones' (1983) grid/triangle method. A point-in-polygon selection, using a bounding quadrilateral, is used to select the raingauges. The user may add further gauges to the resulting list.

The user chooses the period for which data are to be generated. The program then generates a grid and weights, and thus a catchment average rainfall, for each day in the requested period.

3.10.1 How to run CADRE

A number of disks need to be accessed, the passwords being available from Ann Sekulin for MET and CACHE, and from David Boorman for FLOOD.

```
LINK MET 191 281 RR
password: RR...
ACCESS 281 F/A
LINK CACHE 191 282 RR
password: RC...
ACCESS 282 G/A
LINK FLOOD 192 284 RR
password: RC...
ACCESS 284 K
```

The link numbers and access letters used above are examples only.

With the MET and CACHE disks attached it is possible to retrieve data for particular gauges using the METOFF subroutine, but you will need a password (available from Duncan Reed) if you wish to access pre-1961 data.

To use the CADRE program you need to have a bounding quadrilateral defined in ORACLE. If there is not one already there (and CADRE will tell you if there is not), then you will need to log on to ORACLE as FLOOD (password ******), and

```
START POLYGON.SQL.K
```

To run the main program:

```
Ready ....... CADRE
Catchment number: nnnnn
Please wait - retrieveing gauge numbers ......
xxxxx gauges found from search of expanded quad
Number of extra gauges to include: n
( Give gauge numbers: nnnnnn nnnnnn ...     )
RAINFALL GAUGES -
Gauge nnnnnn has data for years  xxxx xxxx xxxx .....
Give start and end years: xxxx xxxx
( nnnnnn values missing in the output dataset)
```

3.10.2 File generated

A single file is generated by CADRE, and it contains one catchment average daily rainfall (in mm) per line, in the format F8.2. The file is generated on channel 4, is therefore by default named *FILE FT04F001 A1*. This file is suitable for input to the MIMIC microcomputer package.

3.10.3 Subroutines used

QUOFF

RNGLIST

FLOGON

LOGOFF

GRDXY

TRIANG

GETMET

SALT

GETYRS

RGSOFF

AROFF

SAROFF

4.0 Stored routines for simple database operations

4.1 Introduction

These routines are stored under three headings

1. SQL*plus for terminal access via the query langauge,
2. SQL*forms for terminal access via interactive form filling, and
3. FORTRAN subroutines for interaction with the data from within programs.

*4.2 SQL*plus*

4.2.1 General

It is very easy to master the basics of SQL*plus so that commands can be constructed to view, enter, update and delete data. The user can have access to tables stored under another user-id provided permission has been granted. The permission can restrict access so that, for example, data can be selected and updated but not deleted. The granting of a restricted permission protects the data, and gives reassurance to users that they cannot inadvertantly delete data. Changes to the database are usually made immediately, but this can be prevented by issuing the command `SET AUTOCOMMIT OFF`. Changes will now only be made when the command `COMMIT` is entered. If you want to undo changes then type `ROLLBACK WORK` and the database is restored to the state it was in when you last typed `COMMIT` or when you logged on. Beware though that if you log off all uncommitted work is committed so rollback first.

Although it is easy to construct basic SQL*plus queries it is worth storing some commands in files either because they are used frequently, or because they are more complex. The following sections describe the stored SQL*plus commands for selecting, updating, and inserting data.

Incidentally at IH we pronounce SQL as 'esskewelle' but many other people say 'sequel'.

4.2.2 SELECT

SELECT commands are the most used within SQL*plus. The basic syntax is SELECT columns FROM tables WHERE conditions. When combined with formatting instructions, neat lineprinter listings can be generated with page headings, titles and footings.

4.2.2.1 CATCHEV (printer)

CATCHEV provides a lineprinter listing of all the events on a nominated catchment. The output is headed with the catchment number and the name of the gauging station. For each event the application, collector/project code, event number and start and end dates and times of the flow record are listed.

To use, log on and link to FLOOD's D-disk, log on to ORACLE, and type

```
START CATCHEV.SQLSP.
```

The catchment number must then be entered but it is not prompted. This exec switches printing of some columns off which may affect subsequent queries.

4.2.2.2 *EVCOMPS (printer)*

EVCOMPS provides a lineprinter listing of all the component data strings for all events on a specified catchment. Each event starts on a new page, so this can produce a very long output listing. The output is headed with the catchment number, the name of the gauging station, the collector/project code and the event number. For each time series component of an event the site, data type, start and end times, and data interval are listed.

To use, log on and link to FLOOD's D-disk, log on to ORACLE, and type

```
START EVCOMPS.SQLSP.
```

The catchment number must then be entered but it is not prompted. This exec switches printing of some columns off which may affect subsequent queries.

4.2.2.3 *WHATRAIN (terminal)*

WHATRAIN lists all the rainfall data for a catchment between specified dates. Since the gauges examined for data are those found in RAIN_GAUGE_LIST, the catchment must have a surrounding quadrilateral defined and must have been run through POLYON. See Section 3.2.

To use, log on and link to FLOOD's D-disk, log on to ORACLE, and type

```
START WHATRAIN.SQL.
```

Enter the catchment number and start and end dates (in format dd-mm-yy) when prompted. This retrieval is very slow.

4.2.2.4 *RAINHOUR (terminal)*

RAINHOUR lists the days for which hourly rainfalls are stored for a particular raingauge.

To use, log on and link to FLOOD's D-disk, log on to ORACLE, and type

```
START RAINHOUR.SQLST.
```

Enter the raingauge number when prompted. Should any day be incomplete then it will still appear in the listing.

4.2.2.5 *LISTS OF RAINGAUGES EX RAINMASTER (printer)*

A straightforward list of raingauges in gauge number order can be obtained showing most of the information held on tables RAIN_GAUGE_SITE and RAIN_GAUGE_DETAIL. As the complete list is very long, a range of raingauges to be printed has to be specified.

To use, log on and link to FLOOD's D-disk, log on to ORACLE, and type

```
START RAINGL.SQLST.
```

Enter the range of raingauge numbers, first the lower number <ENTER>, followed by the higher number <ENTER>. These two numbers are not prompted.

A second listing in gauge number order is available showing only comments and/or start dates and end dates applicable to recording raingauges.

To use, log on and link to FLOOD's D-disk, log on to ORACLE, and type

```
START RAINGC.SQLST.
```

Enter the range of raingauge numbers, first the lower number <ENTER>, followed by the higher number <ENTER>. These two numbers are not prompted.

4.2.3 INSERT

INSERT commands are used within SQL*plus to add data to a table, or, when combined with SELECT, to effect a COPY, this command not existing within SQL*plus.

4.2.3.1 EVCOPY (terminal)

EVCOPY creates a new event identical to a given event except for application and collector codes. Selected components can also be copied across to the new event. The routine sets up an event cross-reference on ORACLE table EVENT_REFERENCE between the old and new event. This SQL*plus routine is particularly useful when dealing with long, complex events during analysis. The complete event can be copied to a new event number under collector code 20 for "rainfall-runoff modelling", whilst the original event may be truncated for FSR-type analysis purposes. The job can be run more than once to create several short events from one long one; when doing this use the original long event as the old event each time.

To use, log on and link to FLOOD's D-disk, log on to ORACLE, and type

```
START EVCOPY.SQL
```

and enter the old event number, new application, new collector code, and the data types and intervals of the components to be copied across to the new event when prompted. The new event number will be printed out at the end of the routine. eg.

```
START EVCOPY.SQL
Enter value for new_application:
1
Enter value for new_collected_by:
20

1 record created.

Enter value for old_event:
4296

1 record updated.

1 record created.

Enter value for data_types_to_be_copied:
1,4,10,13,17
Enter value for data_intervals_to_be_copied:
1,24

12 records created.

NEW_EVENT_NUMBER
----------------
            4359
```

Note that it will not be possible to adjust the rainfall of the truncated event using CARP as the program will now find overlapping events. Component dates must be adjusted as described in EVDEFS2 (Section 4.3.2).

4.2.3.2 Catchment average SMD (CASMD)

If more than one SMD station is to be used, calculate the average SMD for the catchment for the rainfall start date using a simple or weighted average as appropriate. You will need to take into account the position of the gauge in relation to the catchment, its aspect, the type of soil, etc. If only one station value is available, then just use that figure.

Enter these values onto ORACLE table CASMD eg.

```
insert into FLOOD.CASMD (catchment, time, smd)
values (23006,to_date('&smddate 09.00','dd-mm-yy hh24.mi'),&smd);
```

Then enter the required details onto ORACLE table COMPONENT eg.

```
undefine smddate
insert into FLOOD.COMPONENT
(event, site, data_type, begin, until, interval)
values
(&event, 23006, 4, to_date('&&smddate 09.00','dd-mm-yy hh24.mi'),
                   to_date('&smddate 09.00','dd-mm-yy hh24.mi'),24);
```

Note that because only one day is involved, the same date has to be put in both the columns 'begin' and 'until'.

The routine SMDCOMPS.SQLI on FLOOD's D-disk inserts SMD components for a set of events with a particular collector/project code on a nominated catchment, and is convenient to use if you have more than a few events for the same catchment. The start date of the catchment average rainfall for each event is used to calculate the date of the SMD, so this job will not work unless you have run CARP (Section 3.4). In fact, inserting the SMD components is one of the last things you should do before running RUNLPROG (Section 3.6). To use, log on and links to FLOOD's D-disk, log on to ORACLE, and type

```
START SMDCOMPS.SQLI
```

You will be prompted for the catchment number, the collector/project code and the lowest and highest event numbers of those events requiring SMD components eg.

```
START SMDCOMPS.SQLI
Enter value for site:
23006
Enter value for collected_by:
13
Enter value for first_event:
4219
Enter value for last_event:
4223

5 records created.
```

The SMD components which have been loaded onto ORACLE can be checked by logging on to ORACLE and typing the SQL*plus command

```
SELECT * FROM COMPONENT WHERE EVENT=4219;

  EVENT DATA_TYPE    SITE BEGIN     UNTIL     INTERVAL
------- --------- ------- --------- --------- --------
   4219         1   23006 26-OCT-80 30-OCT-80        1
   4219         4   23006 26-OCT-80 26-OCT-80       24
   4219        10   13545 26-OCT-80 28-OCT-80       24
   4219        10   13553 26-OCT-80 .........       ..
   ....        ..  ...... ......... .........       ..
   ....        ..  ...... ......... .........       ..
   4219        10  604039 26-OCT-80 28-OCT-80       24
   4219        13   23006 26-OCT-80 28-OCT-80        1
   4219        17   23006 21-OCT-80 25-OCT-80       24
```

*4.3 SQL*forms*

SQL*forms is the name of a facility that allows forms to be used to see, insert, delete and update data on ORACLE tables. It not only allows pre-defined forms to be used but lets you create new ones to perform new operations. Creating a new form can be a very efficient way of carrying out a database operation. An earlier forms utility called IAP (Interactive Application Processor) is still used and operates in the same way as SQL*forms.

Performing operations on forms requires the use of function keys and is therefore much easier on the colour or 132 column terminals on which all 24 PF keys are single-stroke. You can also use a template to remind you of key functions (see MCC for a copy).

4.3.1 PF key functions

4.3.2 EVDEFS2

New event numbers need to be set for all events that are to be processed further, defining the start and end times of the FLOW in the first instance. This is done by a form filling exercise on the IBM which is easier to do using one of the colour screens in the terminal room. Logon and link to FLOOD's D-disk and type

```
IAP EVDEFS2
```

The program will direct you to log on to ORACLE.

A form will appear on the screen and you will need to use the PF keys to move around it. This is why it would be helpful to have a template for the PF keys giving details of their functions for this specific task.

Press PF11 to go to the next block.

Enter appropriate event details for one event.

```
Event number          99999
Application              01     (for lumped rainfall-runoff
Collector/collator       13     (or whatever project code is applicable
```

Press <ENTER>.

A message will be shown on the screen to the effect that it has failed.

Press PF1 to continue.

The entry will now be for Lumped Rainfall/Runoff and a Post-FSSR16 event.

Press PF6 to commit the work.

A 4-figure event number will be shown. Please write it on the flow form in the file as it will be needed again for reference.

Press PF10 to return to the previous block.

Enter the corresponding flow component details for that event. Input the timing details using the normal 24-hour clock, remembering that midnight is 00.00 on the day following, and that the timing interval is given as 1.0 for 1 hour, 0.3 for half-hour, and 0.15 for quarter-hour. The headings for the columns in ORACLE table COMPONENT with a typical entry are

```
EVENT     TYPE     SITE     BEGIN      TIME   UNTIL      TIME   INT
as given  1(flow)  cat.no.  dd-mon-yy  hh.mm  dd-mon-yy  hh.mm  interval
1234      1        19001    01-jan-81  12.00  03-jan-81  22.00  1.0
```

Press PF6 to commit the work.

Press PF11 to go to the next block.

Press (PF12 + PF14) to create another record and clear the previous details from the form.

Repeat the process from the beginning. etc. This time, when you return to the previous block, input event details at the next record using PF8. Continue in this way until all events for the catchment have been allocated event numbers. It is better for subsequent analyses if the events follow in chronological order, so be sure to get the list of dates sorted out before you start allocating numbers.

When the catchment is completed, or if you just want to stop, press PF3 to quit.

If the start and end dates and times of the flow or other data need subsequent amendment ie. if the event needs to be shortened or lengthened after running CARP and PERCH, then this can also be done through EVDEFS2. Type

```
IAP EVDEFS2
```

Log on to ORACLE as directed.

Press PF4 to enter a query.

Input the required event number and press < ENTER > . The details of the components for that event will be displayed on the screen. Identify the various components by their data types. Data types are described in detail on ORACLE table DATA_TYPE.

Press PF2 to enable the incorrect record to be updated.

Alter the date and times as necessary. Note that because of the way in which the dates and times are stored on ORACLE, it is not possible to change a time without altering a date. If only a time needs to be changed, you must type over part of a date field as well. This fools the computer into thinking that a proper change has occurred, and the record is updated accordingly.

Press PF6 to commit the work.

If there are any more amendments to be done, clear the block using PF17, and enter the next query.

4.4 FORTRAN Subroutines

The following subroutines exist to perform database operations. They are held in the subroutine library (TXTLIB) FLOODSQL, found on user-id FLOOD's D-disk. Other subroutines exist, and may also be in the library FLOODSQL, but are extremely unlikely to be of general use, mainly because they were developed to load data from files dumped from Cache-Cache.

COMMON /DIAG/IPOP The common block DIAG contains the single integer variable IPOP that controls the quantity of diagnostic printing produced by a subroutine. Values range from 0 for no printing to 9 for maximum output.

4.4.1.1 APPOFF FSQLSUB

Returns whole of APPLICATION table.

```
      SUBROUTINE APPOFF(APDESC,NAP,ERROR)

Arguments:
APDESC(20)     CHARACTER*40   Contains application descriptions on exit
NAP            INTEGER        Number of applications
ERROR          INTEGER        0 if OK
                              1 if SQL select error
                              2 if not found
                              3 if cursor open error

Common: None

Used in:
```

4.4.1.2 AROFF FSQLSUB

Returns area of catchment from table SWA.STATION_DETAIL.

```
      SUBROUTINE AROFF(CATCH,AREA,ERROR)

Arguments:
CATCH          INTEGER        Catchment number passed to subroutine
AREA           REAL           Contains area on exit
                              Set to -1 if no area for catchment
ERROR          INTEGER        0 if OK
                              1 if not found
                              2 if SQL select error

Common: None

Used in: CARP, PERCH, LP5, IEM4, CATALOG, CADRE
```

4.4.1.3 CAR3DEL FSQLSUB

Deletes type 3 (CARP) catchment average rainfall data for specified catchment, date/time and interval from table CARAIN3.

```
      SUBROUTINE CAR3DEL(CATCH,DAY,DINTO,STR,ERROR)

Arguments:
CATCH          INTEGER        Catchment number passed to subroutine
DAY            CHARACTER*17   dd-mon-yyyy hh.mi
DINTO          REAL           Data interval (duration)
STR            INTEGER        Stream for output
ERROR          INTEGER        0 if OK
                             -1 if not found (ie. OK so no need to delete)
                              2 if SQL select error
                              3 if SQL delete error

Common: /DIAG/IPOP

Used in: CARP
```

4.4.1.4 CAR3ON FSQLSUB

Inserts type 3 (CARP) catchment average rainfall into table CARAIN3. If it already exists for the catchment, date/time and interval the new data are NOT loaded.

```
      SUBROUTINE CAR3ON(ICATCH,DAY,DINTO,ERAIN,IERROR)
```

Arguments:

ICATCH	INTEGER	Catchment number passed to subroutine
DAY	CHARACTER*17	dd-mon-yyyy hh.mi
DINTO	REAL	Data interval (duration)
ERAIN	REAL	Rainfall in mm
IERROR	INTEGER	0 if OK -1 existing entry differs 1 if SQL select error 2 if not found

Common: /DIAG/IPOP

Used in: CARP, IEM4

4.4.1.5 CASMDOFF FSQLSUB

Returns catchment average SMD values for particular dates.

```
      SUBROUTINE CASMDOFF(CATCH,DATE,CASMD,ERROR)
```

Arguments:

CATCH	INTEGER	Catchment number
DATE	CHARACTER*11	SMD date dd-mon-yyyy
CASMD	REAL	SMD value
ERROR	INTEGER	0 if OK 1 if not found 2 if SQL select error

Common: None

Used by:

4.4.1.6 CATOFF FSQLSUB

Gets ordered list of catchment numbers from table FSSR16_CATCHMENT_RESULT on repeated calling.

```
      SUBROUTINE CATOFF(CATCHMENT,ERROR)
```

Arguments:

CATCHMENT	INTEGER	Catchment number
ERROR	INTEGER	0 if OK 1 if SQL select error 2 if cursor open error 3 if not found

Common: None

Used in:

4.4.1.7 CODEL FSQLSUB

Deletes all COMPONENT table entries for a given event number and data type.

```
      SUBROUTINE CODEL(EVENT,DTYPE,ERROR)
```

Arguments:

EVENT	INTEGER	Event number
DTYPE	INTEGER	Data type code
ERROR	INTEGER	0 if OK 1 if not found 2 if SQL select error

Common: None

Used in: CARP

4.4.1.8 COFFX FSQLSUB

Gets the best event number for the Representative Basin Catalogue (ie. right application code and collector/project code, and if lots are found takes the largest one).

```
      SUBROUTINE COFFX(SITE,DAY,APPLIC,COLLBY,TYPE,EVENT,DINT,ERROR)
```

Arguments:

SITE	INTEGER	Number for site (eg. flow or rain gauge)
DAY	CHARACTER*11	dd-mon-yyyy
APPLIC	INTEGER	Application code
COLLBY	INTEGER	Collector/project code
TYPE	INTEGER	Data type code
EVENT	INTEGER	Event number on exit
DINT	REAL	Data interval on exit
ERROR	INTEGER	0 if OK
		1 if SQL select error
		2 if not found

Common: None

Used by: PERCH, CATALOG

4.4.1.9 COFF1 FSQLSUB

Returns component information given site, start date, application code, collector/project code and data type.

```
      SUBROUTINE COFF1(SITE,DAY,APPLIC,COLLBY,TYPE,EVENT,DINT,ERROR)
```

Arguments:

SITE	INTEGER	Number for site (eg. flow or rain gauge)
DAY	CHARACTER*11	dd-mon-yyyy
APPLIC	INTEGER	Application code
COLLBY	INTEGER	Collector/project code
TYPE	INTEGER	Data type code
EVENT	INTEGER	Event number on exit
DINT	REAL	Data interval on exit
ERROR	INTEGER	0 if OK
		1 if SQL select error
		2 if not found

Common: None

Used by: PERCH, LP5, IEM4, ROFF FSQLSUB

4.4.1.10 COFF10 FSQLSUB

Returns start date of a particular component of an event given event number and data type.

```
      SUBROUTINE COFF10(EVENT,TYPE,SDATE,ERROR)
```

Arguments:

EVENT	INTEGER	Event number
TYPE	INTEGER	Data type code
SDATE	CHARACTER*11	Start date dd-mon-yyyy
ERROR	INTEGER	0 if OK
		1 if SQL select error
		2 if not found

Common: None

Used by: CARP

4.4.1.11 COFF11 FSQLSUB

Returns collector/project code for a particular event.

```
      SUBROUTINE COFF11(EVENT,COLLBY,ERROR)
```

Arguments:

EVENT	INTEGER	Event number
COLLBY	INTEGER	Collector/Project code
EVENT	INTEGER	Event number on exit
ERROR	INTEGER	0 if OK
		1 if SQL select error
		2 if not found

Common: None

Used by: CARP

4.4.1.12 COFF2 FSQLSUB

Returns data interval given site, event number and data type.

```
      SUBROUTINE COFF2(SITE,EVENT,TYPE,DINT,ERROR)
```

Arguments:

SITE	INTEGER	Number for site (eg. flow or rain gauge)
EVENT	INTEGER	Event number
TYPE	INTEGER	Data type code
DINT	REAL	Data interval on exit
ERROR	INTEGER	0 if OK
		1 if SQL select error
		2 if not found

Common: None

Used by: LPS

4.4.1.13 COFF3 FSQLSUB

Retrieves component information for SCHEME. Returns the flow, hourly raingauge and daily raingauge details given event number and reference catchment.

```
      SUBROUTINE COFF3(CATCH,EVENT,STIME,ETIME,ERROR)
```

Arguments:

CATCH	INTEGER	Catchment number
EVENT	INTEGER	Event number
STIME	CHARACTER*17	Start date dd-mon-yyyy hh24.mi
ETIME	CHARACTER*17	End date dd-mon-yyyy hh24.mi
ERROR	INTEGER	0 if OK
		1 if SQL select error
		2 if not found
		5 too many flow gauges
		6 too many hourly raingauges
		7 too many daily raingauges

Common: /DIAG/IPOP
        /S/NRS,RSITE(5),NFS,FSITE(5),NDS,DSITE(50)

Used by: SCHEME, ROFF FSQLSUB

4.4.1.14 COFF4 FSQLSUB

Returns site, start time, end time and data interval given event number and data type (up to 50 retrievals).

Note that dates are returned as 'dd-mon-yyyy' (ie. without times); use subroutine COFF5 for the same component information but with times included in the starts and ends.

```
      SUBROUTINE COFF4(EVENT,TYPE,NS,SITES,SDATE,EDATE,DINT,ERROR)
```

Arguments:

EVENT	INTEGER	Event number
TYPE	INTEGER	Data type code
NS	REAL	Number of sites
SITES(50)	INTEGER	Site numbers
SDATE(50)	CHARACTER*11	Start date dd-mon-yyyy
EDATE(50)	CHARACTER*11	End date dd-mon-yyyy

DINT	REAL	Data interval
ERROR	INTEGER	0 if OK
		1 if SQL select error
		2 if not found

Common: None

Used by:

4.4.1.15 COFF5 FSQLSUB

Returns site, start time, end time and data interval given event number and data type (up to 50 retrievals).

Note that dates are returned as 'dd-mon-yyyy hh24.mi' (ie. with times); use subroutine COFF4 for the same component information but with times excluded from the starts and ends.

```
      SUBROUTINE COFF5(EVENT,TYPE,NS,SITES,SDATE,EDATE,DINT,ERROR)
```

Arguments:

EVENT	INTEGER	Event number
TYPE	INTEGER	Data type code
NS	REAL	Number of sites
SITES(50)	INTEGER	Site numbers
SDATE(50)	CHARACTER*17	Start date dd-mon-yyyy hh24.mi
EDATE(50)	CHARACTER*17	End date dd-mon-yyyy hh24.mi
DINT	REAL	Data interval
ERROR	INTEGER	0 if OK
		1 if SQL select error
		2 if not found

Common: None

Used by:

4.4.1.16 COFF7 FSQLSUB

Returns start and end times given site, event number and data type.

```
      SUBROUTINE COFF7(SITE,EVENT,TYPE,SDATE,EDATE,ERROR)
```

Arguments:

SITE	INTEGER	Site number
EVENT	INTEGER	Event number
TYPE	INTEGER	Data type code
SDATE	CHARACTER*17	Start date dd-mon-yyyy hh24.mi
EDATE	CHARACTER*17	End date dd-mon-yyyy hh24.mi
ERROR	INTEGER	0 if OK
		1 if not found
		2 if SQL select error

Common: None

Used by:

4.4.1.17 COFF8 FSQLSUB

Returns components which overlap the start and end dates given (up to 50 retrievals).

```
      SUBROUTINE COFF8(CATCH,DTYPE,APPL,DSTART,DEND,
                       EVNTS,NE,CSTART,CEND,LAP,ERROR)
```

Arguments:

CATCH	INTEGER	Catchment number
TYPE	INTEGER	Data type code
APPL	INTEGER	Application code
DSTART	CHARACTER*17	Start data dd-mon-yyyy hh24.mi
DEND	CHARACTER*17	End date dd-mon-yyyy hh24.mi
EVNTS(50)	REAL	Overlapping event numbers
NE	INTEGER	Number of events
CSTART(50)	CHARACTER*17	Event start dates dd-mon-yyyy hh24.mi
CEND(50)	CHARACTER*17	Event end dates dd-mon-yyyy hh24.mi
LAP(50)	INTEGER	1 if event within dates given

		2 if event encloses dates given
		3 if event overlaps end date given
		4 if event overlaps start date given
		5 if event with same dates
ERROR	INTEGER	0 if OK
		1 if not found
		2 if SQL select error

Common: /DIAG/IPOP/

Used by:

4.4.1.18 COFF9 FSQLSUB

Returns ordered list of events with catchment average rainfall for a given catchment (up to 50 retrievals).

```
      SUBROUTINE COFF9(SITE,EVENTS,TYPE,DINT,NS,ERROR)
```

Arguments:

SITE	INTEGER	Catchment number
EVENTS(50)	INTEGER	Event numbers
TYPE(50)	INTEGER	Data type codes
DINT(50)	REAL	Data intervals
NS	INTEGER	Number of events
ERROR	INTEGER	0 if OK
		-2 initial value
		-1 if more than 50 retrievals
		1 if SQL select error
		2 if cursor open error
		3 if not found
		4 if less than 50 retrievals

Common: /DIAG/IPOP/

Used by:

4.4.1.19 COLLECT FSQLSUB

This returns description of collected_by code for a given code.

```
      SUBROUTINE COLLECT(KODE,DESCRIPT,ERROR)
```

Arguments:

KODE	INTEGER	Collector/project code
DESCRIPT	CHARACTER*40	Description of code
ERROR	INTEGER	0 if OK
		1 if not found
		2 if SQL select error

Common: None

Used by: CARP, PERCH, CATALOG

4.4.1.20 COMPON FSQLSUB

Inserts an entry into table COMPONENT.

```
      SUBROUTINE COMPON(IEN,ITYPE,ISITE,DSTART,DEND,DINT,PREPS,IERROR)
```

Arguments:

IEN	INTEGER	Event number
ITYPE	INTEGER	Data type code
ISITE	INTEGER	Site number
DSTART	CHARACTER*17	Start date dd-mon-yyyy hh24.mi
DEND	CHARACTER*17	End date dd-mon-yyyy hh24.mi
DINT	REAL	Data interval
PREPS	INTEGER	Set to 1 for diagnostics
ERROR	INTEGER	0 if OK
		-2 if existing component entry disagrees
		-1 if component entry already on table
		1 if SQL select error

Common: None

Used by: CARP

4.4.1.21 CSMDON FSQLSUB

Loads catchment SMD data to table CASMD.

```
      SUBROUTINE CSMDON(CATCH,DAY,SMD,PREPS,IERROR)
```

Arguments:

CATCH	INTEGER	Catchment number
DAY	CHARACTER*17	SMD date dd-mon-yyyy hh24.mi
SMD	REAL	SMD value
PREPS	INTEGER	Set to 1 for diagnostics
IERROR	INTEGER	0 if OK 1 if SQL select error 2 if SQL insert error

Common: None

Used by:

4.4.1.22 CSTOFF FSQLSUB

Retrieves ordered list of coast coordinates from DIG.COAST_XY_GB_250K.

```
      SUBROUTINE CSTOFF(XL,XU,YL,YU,HAREA,SEQNO,XORD,YORD,N,ERROR)
```

Arguments:

XL	INTEGER	Lower limit on x-axis
XU	INTEGER	Upper limit on x-axis
YL	INTEGER	Lower limit on y-axis
YU	INTEGER	Upper limit on y-axis
HAREA(5000)	INTEGER	Hydrometric area number of coordinate
SEQNO(5000)	INTEGER	Sequence number of coordinate
XORD(5000)	INTEGER	X-axis location of coordinate
YORD(5000)	INTEGER	Y-axis location of coordinate
N	INTEGER	Number of coordinates
ERROR	INTEGER	0 if OK 1 if SQL select error 2 if cursor open error 3 if not found

Common: /DIAG/IPOP

Used by: CARP, CATALOG

4.4.1.23 CZONE FSQLSUB

Retrieves zone code for a particular catchment from table SWA.STATION_DETAIL (zone = -1 for Northern Ireland, 30 for the Channel Islands and 0 for everywhere else).

```
      SUBROUTINE CZONE(CATCH,ZONE,ERROR)
```

Arguments:

CATCH	INTEGER	Catchment number
ZONE	INTEGER	Zone code
ERROR	INTEGER	0 if OK 1 if not found 2 if SQL select error

Common: None

Used by: GETQRAIN

4.4.1.24 DTAP FSQLSUB

Returns data requirements for a given application.

```
      SUBROUTINE DTAP(APPLIC,DTREQ,ERROR)
```

Arguments:

APPLIC	INTEGER	Application code
DTREQ(10)	INTEGER	Data requirement codes
ERROR	INTEGER	0 if OK 1 if SQL select error 2 if not found

Common: None

Used by:

4.4.1.25 *DTOFF FSQLSUB*

Returns whole of DATA_TYPE table.

```
      SUBROUTINE DTOFF(DTDESC,NDT,ERROR)
```

Arguments:

DTDESC(10)	CHARACTER*30	Data type descriptions
NDT	INTEGER	Number of data types
ERROR	INTEGER	0 if OK 1 if SQL select error 2 if not found 3 if cursor define error

Common: None

Used by:

4.4.1.26 *EA1OFF FSQLSUB*

Returns API for an event specified by its ORACLE event number. Makes select from view EVENT_CAAPI1.

```
      SUBROUTINE EA1OFF(EVENT,NDATA,DAY,MON,YR,API,ERROR)
```

Arguments:

EVENT	INTEGER	ORACLE event number
NDATA	INTEGER	Number of APIs returned
DAY,MON,YR	INTEGER	API start date (day,month,year)
API(30)	REAL	Returned APIs
ERROR	INTEGER	0 if OK -3 if 30 times read 1 if SQL select error 2 if cursor define error 3 if not found 4 if API found for already existing time

Common: /DIAG/IPOP

Used by: LP5

4.4.1.27 *EA2OFF FSQLSUB*

Returns API for an event specified by its ORACLE event number. Makes select from view EVENT_CAAPI2.

```
      SUBROUTINE EA2OFF(EVENT,NDATA,DAY,MON,YR,API,ERROR)
```

Arguments:

EVENT	INTEGER	ORACLE event number
NDATA	INTEGER	Number of APIs returned
DAY,MON,YR	INTEGER	API start date (day,month,year)
API(30)	REAL	Returned APIs
ERROR	INTEGER	0 if OK -3 if 30 times read 1 if SQL select error 2 if cursor define error 3 if not found 4 if API found for already existing time

Common: /DIAG/IPOP

Used by: LP5

4.4.1.28 EA3OFF FSQLSUB

Returns API for an event specified by its ORACLE event number. Makes select from view EVENT_CAAPI3.

```
SUBROUTINE EA3OFF(EVENT,NDATA,DAY,MON,YR,API,ERROR)
```

Arguments:

EVENT	INTEGER	ORACLE event number
NDATA	INTEGER	Number of APIs returned
DAY,MON,YR	INTEGER	API start date (day,month,year)
API(30)	REAL	Returned APIs
ERROR	INTEGER	0 if OK -3 if 30 times read 1 if SQL select error 2 if cursor define error 3 if not found 4 if API found for already existing time

Common: /DIAG/IPOP

Used by: LP5

4.4.1.29 EFLOFF FSQLSUB

Returns flow for an event specified by its ORACLE event number. Makes select from view EVENT_FLOW.

```
SUBROUTINE EFLOFF(EVENT,MINT,NDATA,TIME,DAY.MON,YR,FLOW,TIMES,ERROR)
```

Arguments:

EVENT	INTEGER	ORACLE event number
MINT	INTEGER	Data interval in minutes
NDATA	INTEGER	Number of flows returned
TIME	REAL	Flow start time (hours,minutes)
DAY,MON,YR	INTEGER	Flow start date (day,month,year)
FLOW(250)	REAL	Returned flows
TIMES(250)	CHARACTER*17	Times of flows
ERROR	INTEGER	0 if OK -3 if 250 times read 1 if SQL select error 2 if cursor define error 3 if not found 4 if flow found for already existing time

Common: /DIAG/IPOP

Used by: LP5

4.4.1.30 ER1OFF FSQLSUB

Returns AUTOSTORM catchment average rainfall for an event specified by its ORACLE event number. Makes select from view EVENT_CARAIN1.

```
SUBROUTINE ER1OFF(CATCH,EVENT,MINT,NDATA,TIME,DAY.MON,YR,RAIN,RBSA9,ERROR)
```

Arguments:

CATCH	INTEGER	Catchment number
EVENT	INTEGER	ORACLE event number
MINT	INTEGER	Data interval in minutes
NDATA	INTEGER	Number of rainfalls returned
TIME	REAL	Rain start time (hours,minutes)
DAY,MON,YR	INTEGER	Rain start date (day,month,year)
RAIN(250)	REAL	Returned rains
RBSA9	REAL	Rain betweem 9 and storm
ERROR	INTEGER	0 if OK -3 if 250 times read 1 if SQL select error 2 if cursor define error 3 if not found 4 if rain found for already existing time

		5 if RBSA9 SQL select error
		6 if no RBSA9 found
		9 if rain data intervals do not correspond

Common: /DIAG/IPOP

Used by: LP5

4.4.1.31 *ER2OFF FSQLSUB*

Returns SCRAP catchment average rainfall for an event specified by its ORACLE event number. Makes select from view EVENT_CARAIN2.

```
SUBROUTINE ER2OFF(CATCH,EVENT,MINT,NDATA,TIME,DAY.MON,YR,RAIN,RBSA9,ERROR)
```

Arguments:

CATCH	INTEGER	Catchment number
EVENT	INTEGER	ORACLE event number
MINT	INTEGER	Data interval in minutes
NDATA	INTEGER	Number of rainfalls returned
TIME	REAL	Rain start time (hours,minutes)
DAY,MON,YR	INTEGER	Rain start date (day,month,year)
RAIN(250)	REAL	Returned rains
RBSA9	REAL	Rain betweem 9 and storm
ERROR	INTEGER	0 if OK
		-3 if 250 times read
		1 if SQL select error
		2 if cursor define error
		3 if not found
		4 if rain found for already existing time
		5 if RBSA9 SQL select error
		6 if no RBSA9 found
		9 if rain data intervals do not correspond

Common: /DIAG/IPOP

Used by: LP5

4.4.1.32 *ER3OFF FSQLSUB*

Returns CARP catchment average rainfall for an event specified by its ORACLE event number. Makes select from view EVENT_CARAIN3.

```
SUBROUTINE ER3OFF(CATCH,EVENT,MINT,NDATA,TIME,DAY.MON,YR,RAIN,RBSA9,ERROR)
```

Arguments:

CATCH	INTEGER	Catchment number
EVENT	INTEGER	ORACLE event number
MINT	INTEGER	Data interval in minutes
NDATA	INTEGER	Number of rainfalls returned
TIME	REAL	Rain start time (hours,minutes)
DAY,MON,YR	INTEGER	Rain start date (day,month,year)
RAIN(250)	REAL	Returned rains
RBSA9	REAL	Rain betweem 9 and storm
ERROR	INTEGER	0 if OK
		-3 if 250 times read
		1 if SQL select error
		2 if cursor define error
		3 if not found
		4 if rain found for already existing time
		5 if RBSA9 SQL select error
		6 if no RBSA9 found
		9 if rain data intervals do not correspond

Common: /DIAG/IPOP

Used by: LP5

4.4.1.33 *ER4OFF FSQLSUB*

Returns RADAR catchment average rainfall for an event specified by its ORACLE event number. Makes select from view EVENT_CARAIN4.

```
      SUBROUTINE ER4OFF(CATCH,EVENT,MINT,NDATA,TIME,DAY.MON,YR,RAIN,RBSA9,ERROR)
```

Arguments:

CATCH	INTEGER	Catchment number
EVENT	INTEGER	ORACLE event number
MINT	INTEGER	Data interval in minutes
NDATA	INTEGER	Number of rainfalls returned
TIME	REAL	Rain start time (hours,minutes)
DAY,MON,YR	INTEGER	Rain start date (day,month,year)
RAIN(250)	REAL	Returned rains
RBSA9	REAL	Rain betweem 9 and storm
ERROR	INTEGER	0 if OK -3 if 250 times read 1 if SQL select error 2 if cursor define error 3 if not found 4 if rain found for already existing time 5 if RBSA9 SQL select error 6 if no RBSA9 found 9 if rain data intervals do not correspond

Common: /DIAG/IPOP

Used by: LP5

4.4.1.34 ESMDON FSQLSUB

Loads gauge SMD data to table ESMD.

```
      SUBROUTINE ESMDON(GAUGE,DAY,SMD,PREPS,IERROR)
```

Arguments:

CATCH	INTEGER	Gauge number
DAY	CHARACTER*17	SMD date dd-mon-yyyy hh24.mi
SMD	REAL	SMD value
PREPS	INTEGER	Set to 1 for diagnostics
IERROR	INTEGER	0 if OK 1 if SQL select error 2 if SQL insert error

Common: None

Used by: ESMDLOAD

4.4.1.35 ESMOFF FSQLSUB

Returns catchment average SMD values for events. Badly named as you might expect it to return gauge values (ESMD) rather than catchment average values (CASMD).

```
      SUBROUTINE ESMOFF(EVENT,NDATA,DAY,MON,YR,SMD,ERROR)
```

Arguments:

EVENT	INTEGER	Event number
NDATA	INTEGER	Number of SMDs
DAY	INTEGER	Day of SMD
MON	INTEGER	Month of SMD
YR	INTEGER	Year of SMD
SMD	REAL	SMD value
ERROR	INTEGER	0 if OK -3 if more than 250 retrievals 1 if SQL select error 2 if cursor define error 3 if not found

Common: /DIAG/IPOP

Used by: LP5

4.4.1.36 EVCOON FSQLSUB

Loads comments relating to events to table EVENT_COMMENT.

```
      SUBROUTINE EVCOON(EVENT,FLOTEX,IERROR)
```

Arguments:

EVENT	INTEGER	Event number
FLOTEX	CHARACTER*80	Comment on event
IERROR	INTEGER	0 if OK
		2 if SQL insert error
		3 if SQL select error

Common: /DIAG/PREPS

Used by:

4.4.1.37 *EVLIST FSQLSUB*

Returns list of events with flow for a given catchment (data type defined within subroutine) (up to 100 retrievals).

```
      SUBROUTINE EVLIST(CATCH,NE,EVENTS,ERROR)
```

Arguments:

CATCH	INTEGER	Catchment number
NE	INTEGER	Number of events
EVENTS(100)	INTEGER	Event numbers
ERROR	INTEGER	0 if OK
		1 if not found
		3 if SQL select error

Common: None

Used by:

4.4.1.38 *EVLON FSQLSUB*

Allocates new ORACLE event numbers to old Cache-Cache event numbers using table EVLIST.

```
      SUBROUTINE EVLON(ICATCH,IEVENT,OEVENT,IFLAG)
```

Arguments:

ICATCH	INTEGER	Catchment number
IEVENT	INTEGER	Old event number (within catchment) from Cache-Cache
OEVENT	INTEGER	New event number on ORACLE
IFLAG	INTEGER	0 if entry found
		-2 if entry not found, entered successfully
		1 if SQL select error from EVLIST
		2 if SQL select error from EVENT
		3 if SQL insert error in EVLIST
		4 if SQL insert error in EVENT

Common: None

Used by:

4.4.1.39 *EVNTON FSQLSUB*

Inserts entry into table EVENT.

```
      SUBROUTINE EVNTON(APPLIC,COLLEC,OEVENT,ERROR)
```

Arguments:

APPLIC	INTEGER	Application code
COLLEC	INTEGER	Collector/project code
OEVENT	INTEGER	ORACLE event number
ERROR	INTEGER	0 if entry not found, entered successfully
		1 if SQL select error
		2 if SQL insert error

Common: None

Used by:

4.4.1.40 *FEST FSQLSUB*

Returns flood estimates for up to 150 catchments given return period and method from table FLOOD_ESTIMATE.

```
      SUBROUTINE FEST(SITES,NS,RP,PEAKS,METH,ERROR)
```

Arguments:

SITES(150)	INTEGER	Catchment numbers
NS	INTEGER	Number of catchments
RP	INTEGER	Return period
PEAKS(150)	REAL	Estimated flood peaks
METHOD	INTEGER	Method code
ERROR	INTEGER	0 if OK 1 if SQL select error 3 if not found

Common: None

Used by:

4.4.1.41 *FESTCM FSQLSUB*

Returns all flood estimates for a given catchment and method from table FLOOD_ESTIMATE.

```
      SUBROUTINE FESTCM(CATCH,METH,PEAKS,ERROR)
```

Arguments:

PEAKS(6)	REAL	Flood peaks
METH	INTEGER	Method code
CATCH	INTEGER	Catchment number
ERROR	INTEGER	0 if OK 1 if SQL select error 3 if not found

Common: /DIAG/IPOP

Used by:

4.4.1.42 *FESTD FSQLSUB*

4.4.1.43 *FESTM FSQLSUB*

Returns the description of a particular method of flood estimation from table FLOOD_ESTIMATE_METHOD.

```
      SUBROUTINE FESTM(METH,DES,ERROR)
```

Arguments:

METH	INTEGER	Method code
DES	CHARACTER*30	Description of method
ERROR	INTEGER	0 if OK 1 if SQL select error 3 if not found

Common: None

Used by:

4.4.1.44 *FESTON FSQLSUB*

Loads flood estimates by various methods onto table FLOOD_ESTIMATE.

```
      SUBROUTINE FESTON(CATCH,RP,PEAK,METHOD,ERROR)
```

Arguments:

CATCH	INTEGER	Catchment number
RP	INTEGER	Return period
PEAK	REAL	Peak flow
METHOD	INTEGER	Method code
ERROR	INTEGER	0 if OK

		-2 if already on table and same -1 if already on table but different 1 if SQL select error 2 if SQL insert error

Common: /DIAG/IPOP

Used by:

4.4.1.45 FLOCON FSQLSUB

Loads comments relating to flow processing to table FLOW_COMMENT.

```
      SUBROUTINE FLOCON(GAUGE,DAY,DAY2,FLOTEX,IERROR)
```

Arguments:

GAUGE	INTEGER	Flow gauge number
DAY	CHARACTER*17	Start date of flow that comment refers to dd-mon-yyyy hh24.mi
DAY2	CHARACTER*17	End date of flow that comment refers to dd-mon-yyyy hh24.mi
FLOTEX	CHARACTER*80	Comment on flow processing
IERROR	INTEGER	0 if OK 2 if SQL select error

Common: None

Used by: LOADFLOW

4.4.1.46 FLOFF FSQLSUB

Returns flow data given gauge number, start and end times.

```
      SUBROUTINE FLOFF(FGAUGE,STIME,ETIME,FLOWS,TIMES,IFAIL)
```

Arguments:

FGAUGE	INTEGER	Flow gauge number
STIME	CHARACTER*17	Start time dd-mon-yyyy hh24.mi
ETIME	CHARACTER*17	End time dd-mon-yyyy hh24.mi
FLOWS(250)	REAL	Flows
TIMES(250)	CHARACTER*17	Times dd-mon-yyyy hh24.mi
IFAIL	INTEGER	0 if OK 1 if more than 250 values 2 if SQL select error 3 if not found

Common: None

Used by:

4.4.1.47 FLOFF2 FSQLSUB

Returns flow data and number of values given gauge number, start and end times.

```
      SUBROUTINE FLOFF2(FGAUGE,STIME,ETIME,N,FLOWS,TIMES,IFAIL)
```

Arguments:

FGAUGE	INTEGER	Flow gauge number
STIME	CHARACTER*17	Start time dd-mon-yyyy hh24.mi
ETIME	CHARACTER*17	End time dd-mon-yyyy hh24.mi
N	INTEGER	Number of flows
FLOWS(250)	REAL	Flows
TIMES(250)	CHARACTER*17	Times dd-mon-yyyy hh24.mi
IFAIL	INTEGER	0 if OK 1 if more than 250 values 2 if SQL select error 3 if not found

Common: None

Used by:

4.4.1.48 *FLOGON FSQLSUB*

Logs on to ORACLE with user-id FLOOD. It is important to check that the log on has been successful before trying to perform any database operations. Use subroutine ULOGON to log on under any other user-id.

```
      SUBROUTINE FLOGON(IERROR)
```

Arguments:

IERROR	INTEGER	0 if OK
		1 if SQL error

Common: /DIAG/IPOP

Used by: All programs that need to access ORACLE via FLOOD user-id.

4.4.1.49 *FLOWON FSQLSUB*

Loads flow data to table FLOW.

```
      SUBROUTINE FLOWON(GAUGE,DAY,FLOW,PREPS,IERROR)
```

Arguments:

GAUGE	INTEGER	Flow gauge number
DAY	CHARACTER*17	Date dd-mon-yyyy hh24.mi
FLOW	REAL	Flow in cumec
PREPS	INTEGER	Set to 1 for diagnostics
IERROR	INTEGER	0 if OK
		1 if SQL select error
		2 if flow already on table

Common: /DIAG/IPOP

Used by: LOADFLOW

4.4.1.50 *FROFF1 FSQLSUB*

Returns flows and catchment average rainfalls for an event specified by its ORACLE event number. Makes select from views EVENT_FLOW, EVENT_CARAINn.

```
      SUBROUTINE FROFF1(EVENT,WHICH,MINT,NDATA,ERROR)
```

Arguments:

EVENT	INTEGER	Event number
WHICH	INTEGER	Type of catchment average rainfall
MINT	INTEGER	(Flow) data interval in minutes
NDATA	INTEGER	Number of values
ERROR	INTEGER	0 if OK
		-3 if 250 times read
		1 if SQL select error
		2 if cursor define error
		3 if not found
		4 if found for already existing time
		9 if data intervals do not correspond

Common: /DIAG/IPOP
        /DATA/FLOW(250),RAIN(250)
        /T/TIMES(250)

FLOW(250)	REAL	Flow values
RAIN(250)	REAL	Rainfall values
TIMES(250)	CHARACTER*17	Times dd-mon-yyyy hr24.mi

Used by: PERCH, CATALOG, IEM4

4.4.1.51 *FROFF9 FSQLSUB*

Returns flows and hourly and daily rainfalls for an event specified by catchment and ORACLE event number. Makes select from tables FLOW, RAIN.

```
      SUBROUTINE FROFF9(CATCH,EVENT,NDATA,NDATAD,ERROR)
```

Arguments:

CATCH	INTEGER	Catchment number
EVENT	INTEGER	Event number
NDATA	INTEGER	Number of values
NDATAD	INTEGER	Number of days of data
ERROR	INTEGER	0 if OK -4 if more than 10 days of data -3 if 250 times read 1 if SQL select error 2 if cursor define error 3 if not found

```
Common: /DIAG/IPOP
        /DATA/FLOWS(5,250),RAINS(5,250)
        /DAILY/DRAINS(50,10)
        /DAYT/DATE(10)
        /T/TIMES(250)
        /S/NRS/RSITE(5),NFS,FSITE(5),NDS,DSITE(50)
```

FLOWS(5,250)	REAL	Flow values
RAINS(5,250)	REAL	Hourly rainfall values
DRAINS(50,10)	REAL	Daily rainfall values
DATE(10)	CHARACTER*11	Days of daily data dd-mon-yyyy
TIMES(250)	CHARACTER*17	Times of hourly data dd-mon-yyyy hr24.mi
NRS	INTEGER	Number of hourly raingauges
RSITE(5)	INTEGER	Hourly raingauge numbers
NFS	INTEGER	Number of flow gauges
FSITE(5)	INTEGER	Flow gauge numbers
NDS	INTEGER	Number of daily raingauges
DSITE(50)	INTEGER	Daily raingauge numbers

Used by: RORB, SCHEME, ROFF FSQLSUB

4.4.1.52 *FSRCCS FSQLSUB*

Retrieves all the catchment characteristics needed for the FSR design flood calculations.

```
      SUBROUTINE FSRCCS(CATCH,AREA,SAAR,M52D,RJEN,SMDBAR,
                        SOIL1,SOIL2,SOIL3,SOIL4,SOIL5,URBAN,
                        MSL,SL1085,STMFRQ,LAKE,ERROR)
```

Arguments:

CATCH	INTEGER	Catchment number
AREA	REAL	Topographic drainage area (sq.km)
SAAR	REAL	Standard annual average rainfall (mm)
M52D	REAL	2-day rainfall of 5-year return period (mm)
RJEN	REAL	Jenkinson's r
SMDBAR	REAL	Mean SMD
SOIL1	REAL	
SOIL2	REAL	
SOIL3	REAL	Fraction of WRAP soil type n
SOIL4	REAL	
SOIL5	REAL	
URBAN	REAL	Urban fraction
MSL	REAL	Main stream length (km)
SL1085	REAL	Channel slope (m/km)
STMFRQ	REAL	Stream frequency (junctions/sq.km)
LAKE	REAL	Fraction of area draining through a significant lake
ERROR	INTEGER	0 if OK 1 if not found 2 if SQL select error

Common: None

Used by:

4.4.1.53 *FSRDAT FSQLSUB*

Retrieves FSR unit hydrograph and losses model parameters.

```
      SUBROUTINE FSRDAT(CATCH,DATE,Q,PR,P,SMD,API,ANTF,
                        TP,QP,NEV,ERROR)
```

Arguments:

CATCH	INTEGER	Catchment number
DATE(100)	CHARACTER*11	Date dd-mon-yyyy
Q(100)	REAL	Peak flow (cumec)
PR(100)	REAL	Percentage runoff
P(100)	REAL	Total rainfall (mm)
SMD(100)	REAL	SMD at start of rain
API(100)	REAL	API
ANTF(100)	REAL	Flow at start of rise
TP(100)	REAL	UH time-to-peak (hours)
QP(100)	REAL	UH peak (cumec/100 sq.km)
NEV	INTEGER	Number of events
ERROR	INTEGER	0 if OK
		-1 if more than 100 retrievals
		1 if SQL select error
		2 if cursor open error

Common: /DIAG/IPOP

Used by:

4.4.1.54 FSRON FSQLSUB

Loads results onto table FSR_RESULT.

```
      SUBROUTINE FSRON(ICATCH,IEVENT,OEVENT,SDATE,RTOTAL,DUR,FLPEAK,
                       RBP,DURBP,LAG,ANTECF,ANSF,SMD,API,FLOWMM,PERC,
                       LOSS,CWI,INITLR,FINLR,PIRF,QPEAK,TPEAK,WHALFP,
                       PREPS,IERROR)
```

Arguments:

ICATCH	INTEGER	Catchment number
IEVENT	INTEGER	Event number used in FSR
OEVENT	INTEGER	Event number used on ORACLE
SDATE	CHARACTER*11	Event start date dd-mon-yyyy
RTOTAL	REAL	Rainfall depth (mm)
DUR	REAL	Rainfall duration (hours)
FLPEAK	REAL	Flow peak (cumec)
RBP	REAL	Rainfall depth before flow peak (mm)
DURBP	REAL	Rainfall duration before flow peak (hours)
LAG	REAL	FSR lag (hours)
ANTECF	REAL	Flow at start of rise (cumec)
ANSF	REAL	Average non-separated flow (cumec)
SMD	REAL	SMD at start of rain (mm)
API	REAL	API (mm)
FLOWMM	REAL	Total flow (mm)
PERC	REAL	
LOSS	REAL	
CWI	REAL	
INITLR	REAL	Initial rain loss rate (mm/hr)
FINLR	REAL	Final rain loss rate (mm/hr)
PIRF	REAL	Peak intensity of rainfall (mm/hr)
QPEAK	REAL	UH peak (cumec/100 sq.km)
TPEAK	REAL	UH time-to-peak (hours)
WHALFP	REAL	UH width at half peak (hours)
PREPS	INTEGER	Set to 1 for diagnostics
IERROR	INTEGER	0 if OK
		1 if SQL select error
		2 if SQL insert error

Common: None

Used by:

4.4.1.55 FSSRON FSQLSUB

Loads results onto table FSSR16_RESULT.

```
      SUBROUTINE FSSRON(ICATCH,IEVENT,OEVENT,SDATE,DATAIN,TIMECO,
                        DURNRA,RAINTO,DRAINB,RAINB4,FSRLAG,FLOWPE,
                        FLOWIN,FLOWAE,FLOWAN,DURNFL,SMDINI,APIINI,
                        PERCRO,LRINIT,LRFINA,FSRQP,FSRTP,FSRW,FSRLP6,
                        PREPS,IERROR)
```

Arguments:

ICATCH	INTEGER	Catchment number
IEVENT	INTEGER	Event number used in FSR and FSSR16
OEVENT	INTEGER	Event number used on ORACLE
SDATE	CHARACTER*11	Event start date dd-mon-yyyy
DATAIN	REAL	
TIMECO	REAL	
DURNRA	REAL	Rainfall duration (hours)
RAINTO	REAL	Rainfall depth (mm)
DRAINB	REAL	Rainfall duration before flow peak (hours)
RAINB4	REAL	Rainfall depth before flow peak (mm)
FSRLAG	REAL	FSR lag (hours)
FLOWPE	REAL	Flow peak (cumec)
FLOWIN	REAL	Flow at start of rise (cumec)
FLOWAE	REAL	
FLOWAN	REAL	Average non-separated flow (cumec)
DURNFL	REAL	
SMDINI	REAL	SMD at start of rain (mm)
APIINI	REAL	API (mm)
PERCRO	REAL	
LRINIT	REAL	Initial rain loss rate (mm/hr)
LRFINA	REAL	Final rain loss rate (mm/hr)
FSRQP	REAL	UH peak (cumec/100 sq.km)
FSRTP	REAL	UH time-to-peak (hours)
FSRW	REAL	UH width at half peak (hours)
FSRLP6	REAL	
PREPS	INTEGER	Set to 1 for diagnostics
IERROR	INTEGER	0 if OK 1 if SQL select error 2 if SQL insert error

Common: None

Used by:

4.4.1.56 *GSOFF FSQLSUB*

Retrieves easting and northing of gauging station from table SWA.STATION_DETAIL.

```
      SUBROUTINE GSOFF(CATCH,EAST,NORTH,ERROR)
```

Arguments:

CATCH	INTEGER	Catchment number
EAST	INTEGER	Easting of gauging station
NORTH	INTEGER	Northing of gauging station
ERROR	INTEGER	0 if OK 1 if not found 2 if SQL select error

Common: None

Used by: CARP, CATALOG

4.4.1.57 *GSSOFF FSQLSUB*

Retrieves eastings and northings of gauging stations in defined area from table SWA.STATION_DETAIL.

```
      SUBROUTINE GSSOFF(XL,XU,YL,YU,XORD,YORD,N,ERROR)
```

Arguments:

XL	INTEGER	Lower limit on x-axis
XU	INTEGER	Upper limit on x-axis
YL	INTEGER	Lower limit on y-axis
YU	INTEGER	Upper limit on y-axis
XORD(50)	INTEGER	X-axis location of gauging station
YORD(50)	INTEGER	Y-axis location of gauging station
N	INTEGER	Number of gauging stations
ERROR	INTEGER	0 if OK 1 if SQL select error 2 if cursor open error 3 if not found

Common: /DIAC/IPOP

Used by: CATALOG

4.4.1.58 KERMIT FSQLSUB

Commits changes (usually updates and inserts) made to the database.

```
      SUBROUTINE KERMIT(IERROR)
```

Arguments:

IERROR	INTEGER	0 if OK 1 if SQL error

Common: /DIAG/IPOP

Used by: All programs that update ORACLE tables.

4.4.1.59 LOGOFF FSQLSUB

Logs off ORACLE from any user-id.

```
      SUBROUTINE LOGOFF(IERROR)
```

Arguments:

ERROR	INTEGER	0 if OK 1 if SQL error

Common: /DIAG/IPOP

Used by: All programs that have accessed ORACLE via any user-id.

4.4.1.60 MAXQOFF FSQLSUB

Retrieves the peak flow of a given event from view EVENT_FLOW.

```
      SUBROUTINE MAXQOFF(EVENT,MAXQ,ERROR)
```

Arguments:

EVENT	INTEGER	Event number
MAXQ	REAL	Peak flow (cumec)
ERROR	INTEGER	0 if OK 1 if not found 2 if SQL select error

Common: None

Used by:

4.4.1.61 MINRON FSQLSUB

This subroutine loads PEPR minute rainfall data onto table MRAIN an hour at a time.

```
      SUBROUTINE MINRON(GAUGE,DAY,HTOT,MRAINS,ERROR)
```

Arguments:

GAUGE	INTEGER	Gauge number
DAY	CHARACTER*17	Date/time dd-mon-yyyy hh24.mi
HTOT	REAL	Rainfall total for the hour
MRAINS	CHARACTER*80	Coded minute rainfalls
ERROR	INTEGER	0 if OK 2 if entry already exists 3 if SQL select error

Common: None

Used by:

4.4.1.62 NUHEV FSQLSUB

Returns the number of UH events processed for a catchment.

```
SUBROUTINR NUHEV(CATCH,NEV,TYPE,ERROR)
```

Arguments:

CATCH	INTEGER	Catchment number
NEV	INTEGER	Number of events processed
TYPE	INTEGER	1 for full UH analysis
		2 for losses only
ERROR	INTEGER	0 if OK
		1 if not found
		2 if SQL select error

Common: /DIAG/IPOP

Used by:

4.4.1.63 PSURCH FSQLSUB

Checks that expanded polygon information has been loaded to table SEARCH_POLY.

```
SUBROUTINE PSURCH(CATCH,FACT,ERROR)
```

Arguments:

CATCH	INTEGER	Catchment number
FACT	REAL	Expansion factor
ERROR	INTEGER	0 if OK
		1 if not found
		2 if SQL select error

Common: None

Used by: CARP, CATALOG

4.4.1.64 QSURCH FSQLSUB

Checks that quadrilateral information (QUADX1) has been loaded to table CATCHMENT_QUAD.

```
SUBROUTINE QSURCH(CATCH,ERROR)
```

Arguments:

CATCH	INTEGER	Catchment number
ERROR	INTEGER	0 if OK
		1 if not found
		2 if SQL select error

Common: None

Used by: CARP, CATALOG

4.4.1.65 QUOFF FSQLSUB

Retrieves quad corners and centroid coordinates for a given catchment from table CATCHMENT_QUAD.

```
SUBROUTINE QUOFF(CATCH,QX1,QY1,QX2,QY2,QX3,QY3,QX4,QY4,CX,CY,ERR)
```

Arguments:

CATCH	INTEGER	Catchment number
QX1	REAL	X-coordinate of first corner of quad
QY1	REAL	Y-coordinate of first corner of quad
QX2	REAL	X coordinate of second corner of quad
QY2	REAL	Y coordinate of second corner of quad
QX3	REAL	X coordinate of third corner of quad
QY3	REAL	Y coordinate of third corner of quad
QX4	REAL	X coordinate of fourth corner of quad
QY4	REAL	Y coordinate of fourth corner of quad
CX	REAL	X coordinate of centroid
CY	REAL	Y coordinate of centroid
ERR	INTEGER	0 if OK

		1 if quad found but no centroid
		2 if not found
		3 if SQL select error

Common: /DIAG/IPOP

Used By: GETQRAIN, CARP, CATALOG, CADRE

4.4.1.66 RAINON FSQLSUB

Loads rainfall data onto table RAIN.

```
      SUBROUTINE RAINON(GAUGE,DAY,DINT,RAIN,PREPS,IERROR)
```

Arguments:

GAUGE	INTEGER	Gauge number
DAY	CHARACTER*17	Date dd-mon-yyyy hr24.mi
DINT	REAL	Duration
RAIN	REAL	Rainfall value
PREPS	INTEGER	Set to 1 for diagnostics
IERROR	INTEGER	0 if OK
		1 if SQL select error
		2 if SQL insert error

Common: None

Used by: GETQRAIN, DRLOAD, HOURLOAD, RAINLOAD

4.4.1.67 RBCACO FSQLSUB

Loads Representative Basin catchment comment data to table RB_CATCHMENT_COMMENT.

```
      SUBROUTINE RBCACO(CATCH,FLOTEX,IERROR)
```

Arguments:

CATCH	INTEGER	Catchment number
FLOTEX	CHARACTER*60	Comment on catchment
IERROR	INTEGER	0 if OK
		1 if
		2 if SQL insert error
		3 if SQL select error

Common: /DIAG/PREPS

Used by:

4.4.1.68 RBLRGS FSQLSUB

Retrieves a list of raingauges for a given catchment within a rectangle defined by the expanded search polygon.

```
      SUBROUTINE RBLRGS(CATCH,FACT,NOGS,GAUGES,EASTS,NORTHS,RECORD,
                        FREQ,ERROR)
```

Arguments:

CATCH	INTEGER	Catchment number
FACT	REAL	Expansion factor
NOGS	INTEGER	Number of raingauges
GAUGES(2000)	INTEGER	Raingauge numbers
EASTS(2000)	INTEGER	Eastings of raingauges
NORTHS(2000)	INTEGER	Northings of raingauges
RECORD(2000)	CHARACTER*6	Type of recording raingauge (null if daily)
FREQ(2000)	CHARACTER*2	Frequency of observation
ERROR	INTEGER	0 if OK
		1 if not found
		2 if SQL select error

Common: None

Used by: CATALOG

4.4.1.69 RBSTCO FSQLSUB

Loads Representative Basin station comment data to table RB_STATION_COMMENT.

```
      SUBROUTINE RBSTCO(CATCH,FLOTEX,IERROR)
```

Arguments:

CATCH	INTEGER	Catchment number
FLOTEX	CHARACTER*60	Comment on station
IERROR	INTEGER	0 if OK
		1 if
		2 if SQL insert error
		3 if SQL select error

Common: /DIAG/IPOP

Used by:

4.4.1.70 RCCON FSQLSUB

4.4.1.71 RCON FSQLSUB

4.4.1.72 REFON FSQLSUB

Loads comments regarding event cross-references to table EVENT_REFERENCE and updates table EVENT accordingly.

```
      SUBROUTINE REFON(IEVENT,REFTEX,IFLAG)
```

Arguments:

IEVENT	INTEGER	Event number
REFTEX	CHARACTER*80	Description of event cross-reference
IFLAG	INTEGER	0 if entry found
		-1 if entry not found, entered successfully
		1 if SQL select error looking up entry
		2 if SQL select error from EVENT_REFERENCE
		3 if SQL insert error in EVENT_REFERENCE
		4 if SQL insert error in EVENT

Common: None

Used by:

4.4.1.73 RGCHK FSQLSUB

Checks a particular raingauge is present on table RAIN_GAUGE_SITE.

```
      SUBROUTINE RGCHK(RGN,GRIDE,GRIDN,SAAR,PREPS,IERROR)
```

Arguments:

RGN	INTEGER	Raingauge number
GRIDE	INTEGER	Easting of raingauge
GRIDN	INTEGER	Northing of raingauge
SAAR	INTEGER	SAAR of raingauge
PREPS	INTEGER	Set to 1 for diagnostics
IERROR	INTEGER	0 if OK
		1 if SQL select error
		2 if raingauge features do not correspond
		3 if not found

Common: None

Used by: DRLOAD, HOURLOAD, RAINLOAD

4.4.1.74 RGC10K FSQLSUB

Returns list of raingauges from RAIN_GAUGE_SITE for a 10km square. The zone code must be specified.

```
      SUBROUTINE RGC10K(EAST,NORTH,NOGS,GAUGES,ZONE,ERROR)
```

Arguments:

EAST	INTEGER	Easting of SW corner of grid square
NORTH	INTEGER	Northing of SW corner of grid square
NOGS	INTEGER	Number of raingauges within 10km square
GAUGES(250)	INTEGER	Raingauge numbers
ZONE	INTEGER	Zone code for grid square : -1 for Northern Ireland, 30 for Channel Islands, 0 for rest of UK
ERROR	INTEGER	0 if OK
		-10 initial value
		1 if not found / SQL select error
		2 if cursor open error

Common: /DIAG/IPOP

Used by: GETQRAIN (supercedes RGS10K)

4.4.1.75 RGLIST FSQLSUB

Retrieves a list of raingauges for a given catchment from a search with a specified factor from view RAINGAUGE_LIST. See RNGLIST.

```
      SUBROUTINE RGLIST(CATCH,FACT,NOGS,GAUGES,EASTS,NORTHS,ERROR)
```

Arguments:

CATCH	INTEGER	Catchment number
FACT	REAL	Expansion factor
NOGS	INTEGER	Number of raingauges
GAUGES(2000)	INTEGER	Raingauge numbers
EASTS(2000)	INTEGER	Eastings of raingauges
NORTHS(2000)	INTEGER	Northings of raingauges
ERROR	INTEGER	0 if OK
		1 if not found / SQL select error
		2 if cursor open error

Common: None

Used by:

4.4.1.76 RGNAME FSQLSUB

Gets raingauge name from table RAIN_GAUGE_DETAIL.

```
      SUBROUTINE RGNAME(SITE,GNAME,ERROR)
```

Arguments:

SITE	INTEGER	Raingauge number
GNAME	CHARACTER*20	Name of raingauge
ERROR	INTEGER	0 if OK
		1 if not found
		2 if SQL select error

Common: None

Used by: CARP

4.4.1.77 RGSOFF FSQLSUB

Returns easting, northing, altitude, and SAAR for a given raingauge from table RAIN_GAUGE_SITE.

```
      SUBROUTINE RGSOFF(G,E,N,S,A,ERROR)
```

Arguments:

G	INTEGER	Raingauge number
E	INTEGER	Easting of raingauge
N	INTEGER	Northing of raingauge
S	INTEGER	SAAR of raingauge (mm)
A	INTEGER	Altitude of raingauge (m)
ERROR	INTEGER	0 if OK
		-2 if not found
		1 if SQL select error

Common: /DIAG/IPOP

Used by: CARP, CADRE

4.4.1.78 RGS10K FSQLSUB

Returns list of raingauges from RAIN_GAUGE_SITE for a 10km square. No zone code is specified.

```
      SUBROUTINE RGS10K(EAST,NORTH,NOGS,GAUGES,ERROR)
```

Arguments:

EAST	INTEGER	Easting of SW corner of grid square
NORTH	INTEGER	Northing of SW corner of grid square
NOGS	INTEGER	Number of raingauges within 10km square
GAUGES(250)	INTEGER	Raingauge numbers
ERROR	INTEGER	0 if OK
		-10 initial value
		1 if not found / SQL select error
		2 if cursor open error

Common: /DIAG/IPOP

Used by: originally used in GETQRAIN, but now superceded by RGS10K

4.4.1.79 RNGLIST FSQLSUB

Retrieves a list of rain gauges for a given catchment from a search with a specified factor from tables RAIN_GAUGE_SITE AND SEARCH_POLY.. Only raingauges with zone code 0 are returned, excluding gauges in Northern Ireland and the Channel Islands). See RGLIST.

```
      SUBROUTINE RNGLIST(CATCH,FACT,NOGS,GAUGES,EASTS,NORTHS,ERROR)
```

Arguments:

CATCH	INTEGER	Catchment number
FACT	REAL	Expansion factor
NOGS	INTEGER	Number of raingauges
GAUGES(2000)	INTEGER	Raingauge numbers
EASTS(2000)	INTEGER	Eastings of raingauges
NORTHS(2000)	INTEGER	Northings of raingauges
ERROR	INTEGER	0 if OK
		1 if not found / SQL select error
		2 if cursor open error

Common: None

Used by: CARP, CADRE

4.4.1.80 RNIOFF FSQLSUB

Returns rainfall values and data interval for a particular raingauge given the start and end times.

```
      SUBROUTINE RNIOFF(RGAG,STIM,ETIM,DURAS,N,RAINS,TIMES,ERR)
```

Arguments:

RGAG	INTEGER	Raingauge number
STIM	CHARACTER*17	Start date/time dd-mon-yyyy hh24.mi
ETIM	CHARACTER*17	End date/time dd-mon-yyyy hh24.mi
DURAS(250)	REAL	Duration of rainfalls
N	INTEGER	Number of rainfalls
RAINS(250)	REAL	Rainfall values
TIMES(250)	CHARACTER*17	Date/time dd-mon-yyyy hh24.mi
ERR	INTEGER	0 if OK
		1 if more than 250 rainfalls
		2 if SQL select error
		3 if not found

Common: None

Used by: CARP

4.4.1.81 RNOFF FSQLSUB

Retrieves rainfall for a particular raingauge given the start and end times and the data interval from table RAIN.

```
      SUBROUTINE RNOFF(RGAG,STIM,ETIM,DURA,N,RAINS,TIMES,ERR)
```

Arguments:

RGAG	INTEGER	Raingauge number
STIM	CHARACTER*17	Start time dd-mon-yyyy hh24.mi
ETIM	CHARACTER*17	End time dd-mon-yyyy hr24.mi
DURA	REAL	Data interval
N	INTEGER	Number of rainfalls
RAINS(250)	REAL	Rainfall values
TIMES(250)	CHARACTER*17	Times of rainfalls dd-mon-yyyy hh24.mi
ERROR	INTEGER	0 if OK
		1 if more than 250 retrievals
		2 if SQL select error
		3 if not found

Common: None

Used by: CARP

4.4.1.82 ROFF FSQLSUB

Returns flows and hourly and daily rainfalls for RORB using subroutines COFF1, COFF3 and FROFF9.

```
      SUBROUTINE ROFF(CATCH,DAY,ERROR)
```

Arguments:

CATCH	INTEGER	Catchment number
DAY	CHARACTER*11	Event start date
ERROR	INTEGER	0 if OK
		1 if ORACLE logon error
		2 if error in COFF1
		3 if error in COFF3

```
Common: /DIAG/IPOP
        /DATA/FLOWS(5,250),RAINS(5,250)
        /DAILY/DRAINS(50,10)
        /DAYT/DATE(10)
        /T/TIMES(250)
        /S/NRS/RSITE(5),NFS,FSITE(5),NDS,DSITE(50)
```

FLOWS(5,250)	REAL	Flow values
RAINS(5,250)	REAL	Hourly rainfall values
DRAINS(50,10)	REAL	Daily rainfall values
DATE(10)	CHARACTER*11	Days of daily data dd-mon-yyyy
TIMES(250)	CHARACTER*17	Times of hourly data dd-mon-yyyy hr24.mi
NRS	INTEGER	Number of hourly raingauges
RSITE(5)	INTEGER	Hourly raingauge numbers
NFS	INTEGER	Number of flow gauges
FSITE(5)	INTEGER	Flow gauge numbers
NDS	INTEGER	Number of daily raingauges
DSITE(50)	INTEGER	Daily raingauge numbers

Used by: RORB

4.4.1.83 RPFON FSQLSUB

Loads results to table EVENT_FLOW_RETURN_PERIOD.

```
      SUBROUTINE RPFON(EVENT,FLOW_RP,PEAK,ERROR)
```

Arguments:

EVENT	INTEGER	Event number
FLOW_RP	REAL	Return period of peak flow
PEAK	REAL	Peak flow (cumec)
ERROR	INTEGER	0 if OK
		-2 initial value
		1 if SQL select error

2 if SQL insert error

Common: None

Used by:

4.4.1.84 RPRON FSQLSUB

Loads results to table EVENT_RAIN_RETURN_PERIOD.

```
      SUBROUTINE RPRON(EVENT,DATA_TYPE,TOT_RAIN_RP,TOT_DUR,TOT_DEPTH,
                       MAX_RAIN_RP,MAX_DUR,MAX_DEPTH,ERROR)
```

Arguments:

EVENT	INTEGER	Event number
DATA_TYPE	INTEGER	Data type code
TOT_RAIN_RP	REAL	Return period of event rainfall
TOT_DUR	REAL	Duration of event rainfall
TOT_DEPTH	REAL	Depth of event rainfall (mm)
MAX_RAIN_RP	REAL	Maximum return period within event
MAX_DUR	REAL	Corresponding duration
MAX_DEPTH	REAL	Corresponding depth (mm)
ERROR	INTEGER	0 if OK -2 initial value 1 if SQL select error 2 if SQL insert error

Common: None

Used by:

4.4.1.85 RROFF FSQLSUB

Returns hourly rainfall values for an event specified by its ORACLE event number.

```
      SUBROUTINE RROFF(EV,NRS,RSITE,NDATA,RAINS,ERROR)
```

Arguments:

EV	INTEGER	Event number
NRS	INTEGER	Number of raingauges
RSITE(10)	INTEGER	Raingauge numbers
NDATA	INTEGER	Number of hourly rainfalls
RAINS(10,250)	REAL	Hourly rainfall values
ERROR	INTEGER	0 if OK -3 if more than 250 rainfalls 1 if SQl select error 2 if cursor define error 3 if not found

Common: /DIAG/IPOP
        /T/TIMES

TIMES(250)	CHARACTER*17	Times of hourly data dd-mon-yyyy hh24.mi

Used by: PERCH, CATALOG

4.4.1.86 SALT FSQLSUB

Retrieves the SAAR and altitude of a raingauge from table RAIN_GAUGE_SITE.

```
      SUBROUTINE SALT(RGAG,SAR,ALTI,ERROR)
```

Arguments:

RGAG	INTEGER	Raingauge number
SAR	CHARACTER*17	SAAR (mm)
ALTI	REAL	Altitude (m)
ERROR	INTEGER	0 if OK 2 if not found 3 if SQL select error

Common: None

Used by: CARP, CADRE

4.4.1.87 SAROFF FSQLSUB

Retrieves SAAR (41-70) for a catchment from table SWA.CATCHMENT_CLIMATE.

```
      SUBROUTINE SAROFF(CATCH,SAAR,ERROR)
```

Arguments:

CATCH	INTEGER	Catchment number
SAAR	REAL	SAAR
ERROR	INTEGER	0 if OK 1 if not found 2 if SQL select error

Common: None

Used by: CARP, CADRE

4.4.1.88 SCAT FSQLSUB

Retrieves catchment data for SCHEME from table SCHEME_CATCHMENT.

```
      SUBROUTINE SCAT(CATCHN,REF,SQ,F1,F2,F3,ET,NTH,CLEN,CSLOP,CBRED,
                      RGH,AR,LEN,SLOP,IMP,URB,GG,ERROR)
```

Arguments:

CATCHN	INTEGER	Catchment number
REF	REAL	
SQ	INTEGER	
F1	INTEGER	
F2	INTEGER	
F3	INTEGER	
ET	REAL	
NTH	REAL	
CLEN	REAL	
CSLOP	REAL	
CBRED	REAL	
RGH	REAL	
AR	REAL	
LEN	REAL	
SLOP	REAL	
IMP	REAL	
URB	REAL	
GG	INTEGER	
ERROR	INTEGER	0 if OK 1 if SQL select error 2 if not found

Common: None

Used by: SCHEME

4.4.1.89 SMDOFF FSQLSUB

Returns catchment average SMD values for particular dates.

```
      SUBROUTINE SMDOFF(CATCH,DATE,CASMD,ERROR)
```

Arguments:

CATCH	INTEGER	Catchment number
DATE	CHARACTER*11	SMD date dd-mon-yyyy
CASMD	REAL	SMD value
ERROR	INTEGER	0 if OK 1 if not found 2 if SQL select error

Common: /DIAG/IPOP

Used by:

4.4.1.90 SOIOFF FSQLSUB

Retrieves soil index, based on WRAP, for a catchment from table SWA.CATCHMENT_LANDFORM.

```
      SUBROUTINE SOIOFF(CATCH,SOIL1,SOIL2,SOIL3,SOIL4,SOIL5,ERROR)
```

Arguments:

CATCH	INTEGER	Catchment number
SOIL1	REAL	
SOIL2	REAL	
SOIL3	REAL	Fraction of WRAP soil type n
SOIL4	REAL	
SOIL5	REAL	
ERROR	INTEGER	0 if OK 1 if not found 2 if SQL select error

Common: None

Used by:

4.4.1.91 SRSDON FSQLSUB

Loads 10km grid square reference and date onto table SEARCH_RAIN_SQUARE_DATE.

```
      SUBROUTINE SRSDON(E,N,DAY,ERROR)
```

Arguments:

E	INTEGER	Easting of SW corner of grid square
N	INTEGER	Northing of SW corner of grid square
DAY	CHARACTER*11	Date dd-mon-yyyy
ERROR	INTEGER	0 if OK 1 if SQL insert error

Common: /DIAG/IPOP

Used by: originally used in GETQRAIN, but now superceded by SRSZON

4.4.1.92 SRSOFF FSQLSUB

Checks to see if a 10 km grid square reference and date are on table SEARCH_RAIN_SQUARE_DATE.

```
      SUBROUTINE SRSOFF(E,N,DAY,ERROR)
```

Arguments:

E	INTEGER	Easting of SW corner of grid square
N	INTEGER	Northing of SW corner of grid square
DAY	CHARACTER*11	Date dd-mon-yyyy
ERROR	INTEGER	0 if OK -1 if not found 1 if SQL select error

Common: /DIAG/IPOP

Used by: originally used in GETQRAIN, but now superceded by SRSZOF

4.4.1.93 SRSZOF FSQLSUB

Checks to see if a 10km grid square reference, zone code and date are on table SEARCH_RAIN_SQUARE_DATE.

```
      SUBROUTINE SRSZOF(E,N,DAY,ZONE,ERROR)
```

Arguments:

E	INTEGER	Easting of SW corner of grid square
N	INTEGER	Northing of SW corner of grid square
DAY	CHARACTER*11	Date dd-mon-yyyy
ZONE	INTEGER	Zone code for grid square : -1 for Northern Ireland, 30 for Channel Islands, 0 for rest

		of UK
ERROR	INTEGER	0 if OK -1 if not found 1 if SQL select error

Common: /DIAG/IPOP

Used by: GETQRAIN (supercedes SRSOFF)

4.4.1.94 SRSZON FSQLSUB

Loads 10km grid square reference, zone code and date onto table SEARCH_RAIN_SQUARE_DATE.

```
      SUBROUTINE SRSZON(E,N,DAY,ZONE,ERROR)
```

Arguments:

E	INTEGER	Easting of SW corner of grid square
N	INTEGER	Northing of SW corner of grid square
DAY	CHARACTER*11	Date dd-mon-yyyy
ZONE	INTEGER	Zone code for grid square : -1 for Northern Ireland, 30 for Channel Islands, 0 for rest of UK
ERROR	INTEGER	0 if OK 1 if SQL insert error

Common: /DIAG/IPOP

Used by: GETQRAIN (supercedes SRSDON)

4.4.1.95 STNOFF FSQLSUB

Returns the river and station names from table SWA.STATION_NAME given the catchment number.
6

```
      SUBROUTINE STNOFF(CATCH,RIVER,LOCAT,ERROR)
```

Arguments:

CATCH	INTEGER	Catchment number
RIVER	CHARACTER*20	Name of watercourse
LOCAT	CHARACTER*30	Location of station
ERROR	INTEGER	0 if OK 1 if not found 2 if SQL select error

Common: None

Used by: CARP, PERCH, LP5, GUPPY, IEM4, CATALOG

4.4.1.96 UHRES FSQLSUB

Retrieves results for FSR unit hydrograph and losses model analysis from table UH_ANALYSIS_RESULT.

```
      SUBROUTINE UHRES(CATCH,DATE,Q,PR,P,SMD,API,ANTF,
                       TP,QP,W,INTVL,NEV,ERROR)
```

Arguments:

CATCH	INTEGER	Catchment number
DATE(100)	CHARACTER*11	Date dd-mon-yyyy
Q(100)	REAL	Peak flow (cumec)
PR(100)	REAL	Percentage runoff
P(100)	REAL	Total rainfall (mm)
SMD(100)	REAL	SMD at start of rain
API(100)	REAL	API
ANTF(100)	REAL	Flow at start of rise
TP(100)	REAL	UH time-to-peak (hours)
QP(100)	REAL	UH peak (cumec/100 sq.km)
W(100)	REAL	Width of UH at half peak (hours)
INTVL(100)	REAL	Data interval of analysis
NEV	INTEGER	Number of events
ERROR	INTEGER	0 if OK -1 if more than 100 retrievals 1 if SQL select error 2 if cursor open error

Common: /DIAG/IPOP

Used by: GUPPY

4.4.1.97 ULOGON FSQLSUB

Logs on to ORACLE with user's own user-id. It is important to check that the log on has been successful before trying to perform any database operations.

```
      SUBROUTINE ULOGON(USER,PASS,ERROR)
```

Arguments:

USER	CHARACTER*16	ORACLE user-id
PASS	CHARACTER*16	Password for user-id
ERROR	INTEGER	0 if OK 1 if SQL error

Common: /DIAG/IPOP

Used by: All subroutines that need to access ORACLE data.

4.4.1.98 URBOFF FSQLSUB

Returns the urban fraction of a given catchment from table SWA.CATCHMENT_LANDFORM.

```
      SUBROUTINE URBOFF(CATCH,URB,ERROR)
```

Arguments:

CATCH	INTEGER	Catchment number
URB	REAL	Urban fraction
ERROR	INTEGER	0 if OK 1 not found 2 if SQL select error

Common: None

Used by: GUPPY

5.0 Data structures

5.1 Introduction

5.2 Tables

Data are held on ORACLE tables. A table has columns with specified names that are used to refer to the data. Columns are of a defined type: number, character and date. Number and character columns have specified sizes that of course cannot be exceeded. Date columns may be entered and retrieved in any accepted format, the most common being dd-mon-yyyy hh24.mi. Each entry is a row in the table. In some tables it is permissable to have entries in which only some columns are filled; in such an entry a column with no value is termed 'NULL'. Some columns must always be filled and have been set up so that entries can only be made when values are defined for these mandatory, termed 'NOT NULL', columns. The only other restriction on the data that can be entered on a table is imposed by any indexes that may exist. The main function of an index is to improve speed of access but it can also be used to ensure key values (or combinations of values) are unique. For example, a table may be restricted so that catchment numbers must be unique, or so that the combination of flow gauge number and time is unique.

The way in which the data are stored is very simple. To perform useful operations with the data it is usually necessary to join tables. There are different type of join, but they are based on the same principle, whereby a column (or several columns) in one table are linked to a column (or several columns) of another table. Complex retrievals can join many tables. This joining of tables is fundamental to the usefulness of ORACLE. By breaking down the stored data into its most basic form, it is possible to recombine it in many different ways. If the basic form has been identified properly it should be possible to construct new retrievals that were not envisaged when the database was designed. It is for this reason that so many simple tables have been created.

The following sections give details of each table, with descriptions of the columns, an indication of those columns that must be filled, and notes on indexes on the table and views in which the table is used.

5.2.1 Flood

5.2.1.1 AGROCLIMATIC

Agroclimatic area data from MAFF Bulletin 34.

Column name	Nulls ?	Type	Remarks
ZONE	NOT NULL	CHAR(3)	Zone number
NAME		CHAR(50)	Zone name
SAAR		NUMBER(4)	Standard annual average rainfall in mm
ALTITUDE		NUMBER(4)	Average altitude of zone in m
PT		NUMBER(3)	Potential annual evapotranspiration in mm
GROWS		NUMBER(3)	Day number for start of growing season
GROWL		NUMBER(3)	Length of growing season in days
GROWPT		NUMBER(3)	Growing season potential transpiration
GROWAT		NUMBER(3)	Growing season actual transpiration
WERAIN		NUMBER(4)	Winter excess rainfall in mm
FCS		NUMBER(3)	Day number of return to field

			capacity
FCL		NUMBER(3)	Duration of field capacity in days

5.2.1.2 *APPLICATION*

Description of applications.

Column name	Nulls ?	Type	Remarks
APPLICATION	NOT NULL	NUMBER(3)	Reference number for application type
REMARK	NOT NULL	CHAR(40)	Description of application
FLOW		NUMBER(3)	Flow data
CARAIN1		NUMBER(3)	Autostorm catchment average rainfall data
CARAIN2		NUMBER(3)	SCRAP catchment average rainfall data
CASMD		NUMBER(3)	Catchment average SMD data
SMD		NUMBER(3)	Individual SMD gauge SMD data
API1		NUMBER(3)	Autostorm API
API2		NUMBER(3)	SCRAP API
DT08		NUMBER(3)	Data type 8 (not set)
DT09		NUMBER(3)	Data type 9 (not set)
RAIN		NUMBER(3)	Individual raingauge data

Valid types of application at present are:

APPLICATION	REMARK
1	Lumped rainfall and runoff
2	Gauged flow and rainfall for RORB
3	Flood Routing

5.2.1.3 *AREAL GROUPING OF RAINGAUGES*

Max and min raingauges for HAs.

Column name	Nulls ?	Type	Remarks
HYDROMETRIC_AREA		NUMBER(4)	Hydrometric area number
FIRST_STATION_NO		NUMBER(6)	Lowest raingauge number in hydrometric area
LAST_STATION_NO		NUMBER(6)	Highest raingauge number in hydrometric area
WATER_AUTHORITY		CHAR(40)	Water Authority controlling hydrometric area

5.2.1.4 *CARAIN1*

Derived rainfall data for catchments by time and duration (Autostorm).

Column name	Nulls ?	Type	Remarks
CATCHMENT	NOT NULL	NUMBER(6)	Catchment number using Surface Water Archive convention
TIME	NOT NULL	DATE	Start time (GMT) of rainfall block
DURATION	NOT NULL	NUMBER(5,2)	Duration in hours of rainfall block
RAIN	NOT NULL	NUMBER(4,1)	Total rainfall depth in mm falling in specified duration

Index: CR1_CTD on columns CATCHMENT, TIME, DURATION

Used in view: EVENT_CARAIN1

5.2.1.5 *CARAIN2*

Derived rainfall data for catchments by time and duration (SCRAP).

Column name	Nulls ?	Type	Remarks
CATCHMENT	NOT NULL	NUMBER(6)	Catchment number using Surface

			Water Archive convention
TIME	NOT NULL	DATE	Start time (GMT) of rainfall block
DURATION	NOT NULL	NUMBER(5,2)	Duration in hours of rainfall block
RAIN	NOT NULL	NUMBER(4,1)	Total rainfall depth in mm falling in specified duration

Index: CR2_CTD on columns CATCHMENT, TIME, DURATION

Used in view: EVENT_CARAIN2

5.2.1.6 CARAIN3

Derived rainfall data for catchments by time and duration (CARP).

Column name	Nulls ?	Type	Remarks
CATCHMENT	NOT NULL	NUMBER(6)	Catchment number using Surface Water Archive convention
TIME	NOT NULL	DATE	Start time (GMT) of rainfall block
DURATION	NOT NULL	NUMBER(5,2)	Duration in hours of rainfall block
RAIN	NOT NULL	NUMBER(4,1)	Total rainfall depth in mm falling in specified duration

Index: CR3_CTD on columns CATCHMENT, TIME, DURATION

Used in view: EVENT_CARAIN3

5.2.1.7 CARAIN4

Derived rainfall data for catchments from weather radar.

Column name	Nulls ?	Type	Remarks
CATCHMENT	NOT NULL	NUMBER(6)	Catchment number using Surface Water Archive convention
TIME	NOT NULL	DATE	Start time (GMT) of rainfall block
DURATION	NOT NULL	NUMBER(5,2)	Duration in hours of rainfall block
RAIN	NOT NULL	NUMBER(4,1)	Total rainfall depth in mm falling in specified duration

Index: CR4_CTD on columns CATCHMENT, TIME, DURATION

Used in view: EVENT_CARAIN4

5.2.1.8 CASMD

Derived soil moisture deficit data for catchments by time.

Column name	Nulls ?	Type	Remarks
CATCHMENT	NOT NULL	NUMBER(6)	Catchment number using Surface Water Archive convention
TIME	NOT NULL	DATE	Time (GMT) of SMD
SMD	NOT NULL	NUMBER(4,1)	Soil moisture deficit in mm

Index: CSMD_CT on columns CATCHMENT, TIME

Used in view: EVENT_CASMD

5.2.1.9 CATCHMENT NUMBER UPDATE

Table containing old and new numbers of catchments which have been renumbered, together with the date of update of other ORACLE tables using SQL*plus routine CATNOUP.SQL.

Column name	Nulls ?	Type	Remarks
OLD_NO	NOT NULL	NUMBER(6)	Previous catchment number (a catchment may have more than one entry in this column)
NEW_NO	NOT NULL	NUMBER(6)	Present catchment number
CHANGE_DATE	NOT NULL	DATE	Date and time of update of other ORACLE tables

5.2.1.10 CATCHMENT QUAD

Catchment quad and centroid grid references (new version of table CATCHMENT_DETAILS renamed CATCHMENT_DETAILS_OLD modified August 1990).

Column name	Nulls ?	Type	Remarks
CATCHMENT	NOT NULL	NUMBER(6)	Catchment number using Surface Water Archive convention
CENTREX		NUMBER(5)	X coordinate of centroid
CENTREY		NUMBER(5)	Y coordinate of centroid
QGRID		NUMBER(3)	Grid code for quad
QUADX1		NUMBER(5)	X coordinate of first corner of quad
QUADY1		NUMBER(5)	Y coordinate of first corner of quad
QUADX2		NUMBER(5)	X coordinate of second corner of quad
QUADY2		NUMBER(5)	Y coordinate of second corner of quad
QUADX3		NUMBER(5)	X coordinate cf third corner of quad
QUADY3		NUMBER(5)	Y coordinate of third corner of quad
QUADX4		NUMBER(5)	X coordinate of fourth corner of quad
QUADY4		NUMBER(5)	Y coordinate of fourth corner of quad

Index: CATCH_Q on column CATCHMENT

5.2.1.11 COLLECTOR

Table of projects collecting event data.

Column name	Nulls ?	Type	Remarks
CODE	NOT NULL	NUMBER(3)	Code number for collecting project
COLLECTOR	NOT NULL	CHAR(40)	Description of collecting project

Valid codes and corresponding collector projects are:

CODE	COLLECTOR
10	FSR Or Follow-Up (Replace With 11-13)
11	FSR Original Event
12	FSSR16 Event
13	Post-FSSR16 Event
20	Rainfall-Runoff model studies (Complex)
40	Land Use Change
51	British Coal - Erewash Study
52	Thames Model - hydrological input study
53	AWA Lincoln - SMD investigation
54	North of Scotland Hydro-Electric Board
55	Soar Flood Study
56	Kielder PMF Project
57	Tay Flood Study

5.2.1.12 COMPONENT

Table containing references to all component data required for an event.

Column name	Nulls ?	Type	Remarks
EVENT	NOT NULL	NUMBER(4)	Identifier for event requiring component
DATA_TYPE	NOT NULL	NUMBER(2)	Code to name of table containing data
SITE	NOT NULL	NUMBER(6)	Site number eg. catchment number, raingauge number
BEGIN	NOT NULL	DATE	Start date-time of required data
UNTIL	NOT NULL	DATE	End date-time of required data
INTERVAL	NOT NULL	NUMBER(4,2)	Data interval or block duration for

data in hours

Index: CO_ALL on columns EVENT, DATA_TYPE, SITE, BEGIN, UNTIL, INTERVAL
CO_SI_BE on columns SITE, BEGIN

Used in views: EVENT_CAAPIn, EVENT_CARAINn, EVENT_CASMD, EVENT_DRAIN, EVENT_FLOW, EVENT_HRAIN, EVENT_RAIN, NSHB_RESULT, UH_ANALYSIS_RESULT

5.2.1.13 DATA_TYPE

Table names of data types for each type number.

Column name	Nulls ?	Type	Remarks
CODE	NOT NULL	NUMBER(2)	Data code number
DESCRIPTION	NOT NULL	CHAR(30)	Name of data type

Currently defined data types are:

CODE	DESCRIPTION
1	Flow
2	Catchment rainfall (Autostorm)
3	Catchment rainfall (SCRAP)
4	Catchment SMD from site ESMDs
5	SMD at Met. Office ESMD site
6	Typical gauge API (Autostorm)
7	Average gauge API (SCRAP)
10	Rainfall
13	Catchment rainfall (CARP)
17	Average gauge API (CARP)
99	Catchment rainfall (Radar)

5.2.1.14 ESMD

Estimated soil moisture deficit data for specified sites and times.

Column name	Nulls ?	Type	Remarks
GAUGE	NOT NULL	NUMBER(6)	Raingauge number for Met. Office SMD site
TIME	NOT NULL	DATE	Time (GMT) of SMD
SMD	NOT NULL	NUMBER(4,1)	Soil moisture deficit in mm

Index: GT_ESMD on columns GAUGE, TIME

5.2.1.15 EVENT

Table of event identifiers and their applications.

Column name	Nulls ?	Type	Remarks
EVENT	NOT NULL	NUMBER(4)	Event reference number
APPLICATION	NOT NULL	NUMBER(3)	Application identifier for event
COLLECTED_BY		NUMBER(3)	Code giving project collecting or collating data
REFERENCE		NUMBER(4)	Reference to name/description that links events
REMARK		CHAR(240)	Text remark relating to event processing or analysis

Index: EV_EV on column EVENT

Used in views: EVENT_CAAPIn, EVENT_CARAINn, EVENT_CASMD, EVENT_DRAIN, EVENT_FLOW, EVENT_HRAIN, EVENT_RAIN

5.2.1.16 EVENT COMMENT

Comments relating to event or analysis.

Column name	Nulls ?	Type	Remarks

EVENT	NOT NULL	NUMBER(6)	Event reference number
SEQUENCE	NOT NULL	NUMBER(2)	Sequence number of comment on event
TEXT	NOT NULL	CHAR(80)	Comment on event

Index: EV_SEQ on columns EVENT, SEQUENCE

5.2.1.17 EVENT FLOW RETURN PERIOD

Table containing flow return periods.

Column name	Nulls ?	Type	Remarks
EVENT	NOT NULL	NUMBER(5)	Event reference number
FLOW_RP	NOT NULL	NUMBER(7,2)	Return period in years of peak flow value for event
PEAK	NOT NULL	NUMBER(7,2)	Peak flow value in cumec for event

Index: EVENTRP_F on column EVENT

5.2.1.18 EVENT QUALITY

Table containing quality codes from event analysis.

Column name	Nulls ?	Type	Remarks
EVENT	NOT NULL	NUMBER(4)	Event number on ORACLE
QUALITY		NUMBER(1)	Quality code for event

Valid codes and their meanings at present are:

QUALITY	MEANING
1	Unit Hydrograph event (Tp,SPR)
2	Losses only (SPR)
3	Unused
4	Rejected
5	Snow
	None

5.2.1.19 EVENT RAIN RETURN PERIOD

Table containing rainfall return periods and associated information for events on FSSR16 catchments.

Column name	Nulls ?	Type	Remarks
EVENT	NOT NULL	NUMBER(5)	Event reference number
DATA_TYPE	NOT NULL	NUMBER(2)	Code to name of table containing data
TOT_RAIN_RP	NOT NULL	NUMBER(7,2)	Return period in years of rainfall for whole event
TOT_DUR	NOT NULL	NUMBER(7,2)	Duration in hours of rainfall for whole event
TOT_DEPTH	NOT NULL	NUMBER(7,2)	Depth in mm of rainfall for whole event
MAX_RAIN_RP	NOT NULL	NUMBER(7,2)	Maximum return period in years of rainfall within event
MAX_DUR	NOT NULL	NUMBER(7,2)	Duration in hours of rainfall corresponding to maximum return period
MAX_DEPTH	NOT NULL	NUMBER(7,2)	Depth in mm of rainfall corresponding to maximum return period

Index: EVENTRP_R on columns EVENT, DATA_TYPE

5.2.1.20 EVENT REFERENCE

Column name	Nulls ?	Type	Remarks
REFERENCE	NOT NULL	NUMBER(6)	Reference number of event

			cross-reference
TEXT	NOT NULL	CHAR(80)	Description of event cross-reference

5.2.1.21 *EVLIST*

Table of old event numbers to allocate new event numbers.

Column name	Nulls ?	Type	Remarks
CATCHMENT	NOT NULL	NUMBER(6)	Catchment number using Surface Water Archive convention
EVENT_NUMBER	NOT NULL	NUMBER(3)	Old event number (within catchment) from Cache-Cache
EVENT	NOT NULL	NUMBER(4)	Event number on ORACLE

Index: EL_EV on column EVENT
EL_CE on columns CATCHMENT, EVENT_NUMBER

5.2.1.22 *FESTFITS*

Column name	Nulls ?	Type	Remarks
CATCHMENT	NOT NULL	NUMBER(6)	Catchment number using Surface Water Archive convention
GEV_U		NUMBER(6,2)	Location parameter of GEV distibution
GEV_A		NUMBER(6,2)	Scale parameter of GEV distibution
GEV_K		NUMBER(6,2)	Shape parameter of GEV distibution
MAD		NUMBER(5,3)	
TMAX100		NUMBER(5,1)	Return period at which sd>10.0%
TMAX125		NUMBER(5,1)	Return period at which sd>12.5%
TMAX150		NUMBER(5,1)	Return period at which sd>15.0%

Used in view: FLOOD_ESTIMATE_ANMAX_DETAILS

5.2.1.23 *FLOOD ESTIMATE*

FSR flood estimates and quantiles estimates from data.

Column name	Nulls ?	Type	Remarks
CATCHMENT	NOT NULL	NUMBER(6)	Catchment number using Surface Water Archive convention
RETURN_PERIOD	NOT NULL	NUMBER(4)	Return period at which flood estimate is made
PEAK	NOT NULL	NUMBER(8,2)	Flood estimate at that return period for that catchment
METHOD	NOT NULL	NUMBER(2)	Code to method used to make flood estimate
METHOD_DETAIL		NUMBER(20)	
METHOD_TEXT		CHAR(40)	

Index: FE_CMF on columns CATCHMENT, METHOD, RETURN_PERIOD
FE_MCF on columns METHOD, CATCHMENT, RETURN_PERIOD
FE_FCM on columns RETURN_PERIOD, CATCHMENT, METHOD

Used in views: FLOOD_ESTIMATE_BY_FREQUENCY, FLOOD_ESTIMATE_BY_METHOD

5.2.1.24 *FLOOD ESTIMATE ANMAX ORIGINAL*

Column name	Nulls ?	Type	Remarks
CATCHMENT	NOT NULL	NUMBER(6)	Catchment number using Surface Water Archive convention
FIRST_YEAR		NUMBER(4)	
LAST_YEAR		NUMBER(4)	
YEARS_OF_DATA		NUMBER(3)	
FIT_METHOD		CHAR(3)	
MAX_RETURN_PERIOD		NUMBER(3)	
QUALITY_EXTRAP		NUMBER(2)	

Used in view: FLOOD_ESTIMATE_ANMAX_DETAILS

5.2.1.25 FLOOD ESTIMATE METHOD

Explanation of flood estimate methods.

Column name	Nulls ?	Type	Remarks
METHOD	NOT NULL	NUMBER(2)	Code to method used to make flood estimates
DESCRIPTION	NOT NULL	CHAR(24)	Description of flood estimate method

5.2.1.26 FLOW

Instantaneous flow data corresponding to specified gauges and times.

Column name	Nulls ?	Type	Remarks
GAUGE	NOT NULL	NUMBER(6)	Flow gauge number using Surface Water Archive convention
TIME	NOT NULL	DATE	Time (GMT) of flow
FLOW	NOT NULL	NUMBER(8,4)	Instantaneous flow in cumec

Index: FL_T on column TIME
       FL_GT on columns GAUGE, TIME

Used in view: EVENT_FLOW

5.2.1.27 FLOW COMMENT

Comments relating to flow processing.

Column name	Nulls ?	Type	Remarks
GAUGE	NOT NULL	NUMBER(6)	Flow gauge number using Surface Water Archive convention
BEGIN	NOT NULL	DATE	Start date of flow that comment refers to
UNTIL		DATE	End date of flow that comment refers to
TEXT	NOT NULL	CHAR(80)	Comment on flow processing

5.2.1.28 FSR RESULT

Table containing results on event analysis for FSR.

Column name	Nulls ?	Type	Remarks
EVENT		NUMBER(4)	Event reference number
CATCHMENT	NOT NULL	NUMBER(6)	Catchment number using Surface Water Archive convention
EVENT_NUMBER	NOT NULL	NUMBER(2)	Event numbers used in FSR
STARTING		DATE	Event start date
RTOTAL		NUMBER(4,1)	Rainfall depth in mm
DUR		NUMBER(5,2)	Duration of rainfall in hours
FLPEAK		NUMBER(5,2)	Flow peak in cumec
RBP		NUMBER(4,1)	Rainfall depth before the flow peak in mm
DURBP		NUMBER(5,2)	Duration of rainfall before the flow peak in hours
LAG		NUMBER(5,2)	FSR lag in hours
ANTECF		NUMBER(5,2)	Flow at start of hydrograph rise in cumec
ANSF		NUMBER(5,2)	Average non-separated flow in cumec
SMD		NUMBER(4,1)	Soil moisture deficit at start of rainfall in mm
API		NUMBER(4,1)	Antecedent precipitation index at start of rainfall in mm
FLOWMM		NUMBER(4,1)	Total flow in mm
INITLR		NUMBER(4,1)	Initial rainfall loss rate in mm/hr

FINLR		NUMBER(4,1)	Final rainfall loss rate in mm/hr
PIRF		NUMBER(4,1)	Peak intensity rainfall in mm/hr
QPEAK		NUMBER(5,2)	Unit hydrograph peak in cumec/100 sq.km
TPEAK		NUMBER(5,2)	Unit hydrograph time to peak in hours
WHALFP		NUMBER(5,2)	Unit hydrograph width at half the peak value in hours

```
Index: FSR_E on column EVENT
       FSR_CE on columns CATCHMENT, EVENT_NUMBER
```

5.2.1.29 FSSR16 CATCHMENT RESULT

Table containing catchment results for FSSR16.

Column name	Nulls ?	Type	Remarks
CATCHMENT	NOT NULL	NUMBER(6)	Catchment number using Surface Water Archive convention
TPEAK		NUMBER(4,2)	Time to peak of 1-hour uh in hours
QPEAK		NUMBER(5,1)	Peak of 1-hour uh in cumec/100 sq.km
UH_NUMBER		NUMBER(2)	Number of unit hydrograph events
SPR		NUMBER(4,2)	Standard percentage runoff
PR		NUMBER(4,2)	Percentage runoff
PR_NUMBER		NUMBER(2)	Number of percentage runoff events

5.2.1.30 FSSR16 CODE

Table containing codes from event analysis for FSSR16 (new version of table FSR_CODE renamed August 1990).

Column name	Nulls ?	Type	Remarks
CATCHMENT	NOT NULL	NUMBER(6)	Catchment number using Surface Water Archive convention
EVENT_NUMBER	NOT NULL	NUMBER(2)	Old event number (within catchment) from Cache-Cache
TIMING	NOT NULL	NUMBER(4)	
FLOWS_START	NOT NULL	NUMBER(3)	
FLOWS_END	NOT NULL	NUMBER(3)	
COLLECT	NOT NULL	NUMBER(2)	
QUALITY		NUMBER(1)	
QUAL1		NUMBER(1)	
QUAL2		NUMBER(1)	
QUAL3		NUMBER(1)	
QUAL4		NUMBER(1)	

```
Index: CE_FC on columns CATCHMENT, EVENT_NUMBER
```

5.2.1.31 FSSR16 RESULT

Table containing results on event analysis for FSSR16.

Column name	Nulls ?	Type	Remarks
EVENT	NOT NULL	NUMBER(4)	Event reference number
CATCHMENT	NOT NULL	NUMBER(6)	Catchment number using Surface Water Archive convention
EVENT_NUMBER	NOT NULL	NUMBER(2)	Event numbers used in FSR and FSSR16
STARTING		DATE	Event start date
INTERVAL		NUMBER(3,2)	
TIME_ADJ		NUMBER(4,2)	
RTOTAL		NUMBER(4,1)	Rainfall depth in mm
DUR		NUMBER(5,2)	Duration of rainfall in hours
FLPEAK		NUMBER(5,2)	Flow peak in cumec
RBP		NUMBER(4,1)	Rainfall depth before the flow peak in mm
DURBP		NUMBER(5,2)	Duration of rainfall before the flow peak in hours
LAG		NUMBER(5,2)	FSR lag in hours
ANTECF		NUMBER(5,2)	Flow at start of hydrograph rise in

			cumec
ANSF		NUMBER(5,2)	Average non-separated flow in cumec
BFINC		NUMBER(5,2)	
DURFLOW		NUMBER(5,2)	
SMD		NUMBER(4,1)	Soil moisture deficit at start of rainfall in mm
API		NUMBER(4,1)	Antecedent precipitation index at start of rainfall in mm
PR		NUMBER(4,1)	
INITLR		NUMBER(4,1)	Initial rainfall loss rate in mm/hr
FINLR		NUMBER(4,1)	Final rainfall loss rate in mm/hr
QPEAK		NUMBER(5,2)	Unit hydrograph peak in cumec/100 sq.km
TPEAK		NUMBER(5,2)	Unit hydrograph time to peak in hours
WHALFP		NUMBER(5,2)	Unit hydrograph width at half the peak value in hours
BASE		NUMBER(5,2)	

Index: FSSR16_E on column EVENT
       FSSR16_CE on columns CATCHMENT, EVENT_NUMBER
       CE_F16 on columns CATCHMENT, EVENT_NUMBER

Used in views: NSHB_RESULT, UH_ANALYSIS_RESULT

5.2.1.32 MINUTE RAIN INFO

Table containing name and Met. Office number of gauges on MRAIN.

Column name	Nulls ?	Type	Remarks
MRAIN_CODE	NOT NULL	NUMBER(3)	Code number on MRAIN
SITE_NAME	NOT NULL	CHAR(20)	Name of raingauge site
MET_NUMBER	NOT NULL	NUMBER(8)	Met. Office reference number
GRID_REF	NOT NULL	NUMBER(9)	National grid reference of rain gauge
PERIOD		CHAR(20)	Period of record on MRAIN

5.2.1.33 MINUTE RAINGAUGE SITE

Table containing name and Met. Office number of gauges on MRAIN.

Column name	Nulls ?	Type	Remarks
MRAIN_CODE	NOT NULL	NUMBER(3)	Code number on MRAIN
SITE_NAME	NOT NULL	CHAR(20)	Name of raingauge site
MET_NUMBER	NOT NULL	NUMBER(8)	Met. Office reference number

5.2.1.34 MORECS MONTHLY

MORECS climatology monthly data for 190 40km grid squares (rec 21/8/86). All values are for medium AWC soil (grass).

Column name	Nulls ?	Type	Remarks
YEAR	NOT NULL	NUMBER(4)	Year of record
MONTH	NOT NULL	NUMBER(2)	Month of record
SQUARE	NOT NULL	NUMBER(3)	MORECS square 1 to 190 (40km grid)
PE		NUMBER(4)	Monthly total potential evaporation x10 mm
AE		NUMBER(4)	Monthly total actual evaporation x10 mm
SMD		NUMBER(4)	End of month soil moisture deficit x10 mm
EP		NUMBER(4)	Monthly total effective precipitation x10 mm
RAIN		NUMBER(4)	Monthly total rainfall x10 mm
SUN		NUMBER(4)	Monthly total sunshine x10 hrs
TEMP		NUMBER(4)	Daily mean temperature for month x10 C
VP		NUMBER(4)	Daily mean vapour pressure for

			month x10
WIND		NUMBER(4)	Daily mean wind for month x10 miles/day

Index: MORECS_MONTHLY_INDEX on columns SQUARE, YEAR, MONTH
       MORECS_MONTHLY_INDEX2 on columns YEAR, MONTH, SQUARE

5.2.1.35 *MORECS MONTHLY MEAN*

MORECS climatology monthly data for 190 40km grid squares (last update 04 07 91). All values are for medium AWC soil (grass) and averaged over the period

Column name	Nulls ?	Type	Remarks
START_YEAR	NOT NULL	NUMBER(4)	Start year of record
END_YEAR	NOT NULL	NUMBER(4)	End year of record
MONTH	NOT NULL	NUMBER(2)	Month of record
SQUARE	NOT NULL	NUMBER(3)	MORECS square 1 to 190 (40km grid)
PE		NUMBER(4)	Monthly total potential evaporation x10 mm
AE		NUMBER(4)	Monthly total actual evaporation x10 mm
SMD		NUMBER(4)	End of month soil moisture deficit x10 mm
EP		NUMBER(4)	Monthly total effective precipitation x10 mm
RAIN		NUMBER(4)	Monthly total rainfall x10 mm
SUN		NUMBER(4)	Monthly total sunshine x10 hrs
TEMP		NUMBER(4)	Daily mean temperature for month x10 C
VP		NUMBER(4)	Daily mean vapour pressure for month x10
WIND		NUMBER(4)	Daily mean wind for month x10 miles/day

5.2.1.36 *MRAIN*

Table containing minute rainfalls for 5 UK stations (PEPR data).

Column name	Nulls ?	Type	Remarks
GAUGE	NOT NULL	NUMBER(3)	Name of raingauge site
DATE_TIME	NOT NULL	DATE	Date/time at beginning of hour
HOUR_TOTAL	NOT NULL	NUMBER(6,2)	Total rainfall in hour beginning date_time
MINUTE_RAIN		CHAR(90)	Ebcdic character coding for each minute of rainfall

Index: MR_GT on columns GAUGE, DATE_TIME
       MR_T on columns DATE_TIME, GAUGE

5.2.1.37 *OPERATOR*

Hydrometric network operators.

Column name	Nulls ?	Type	Remarks
CODE	NOT NULL	NUMBER(3)	Code number for operator
NAME	NOT NULL	CHAR(40)	Name of gauge operating authority

Index: OP_C on column CODE

5.2.1.38 *POST FSSR16 RESULT*

Table containing results on event analysis post FSSR16.

Column name	Nulls ?	Type	Remarks
EVENT	NOT NULL	NUMBER(4)	Event reference number
DURFLOW		NUMBER(5,2)	
INTERVAL		NUMBER(3,2)	
DUR		NUMBER(5,2)	Duration of rainfall in hours

RTOTAL		NUMBER(4,1)	Rainfall depth in mm
FLPEAK		NUMBER(5,2)	Flow peak in cumec
LAG		NUMBER(5,2)	FSR lag in hours
ANTECF		NUMBER(5,2)	Flow at start of hydrograph rise in cumec
BFINC		NUMBER(5,2)	
ANSF		NUMBER(5,2)	Average non-separated flow in cumec
PR		NUMBER(4,1)	
INITLR		NUMBER(4,1)	Initial rainfall loss rate in mm/hr
FINLR		NUMBER(4,1)	Final rainfall loss rate in mm/hr
SMD		NUMBER(4,1)	Soil moisture deficit at start of rainfall in mm
API		NUMBER(4,1)	Antecedent precipitation index at start of rainfall in mm
QPEAK		NUMBER(5,2)	Unit hydrograph peak in cumec/100 sq.km
TPEAK		NUMBER(5,2)	Unit hydrograph time to peak in hours
WHALFP		NUMBER(5,2)	Unit hydrograph width at half the peak value in hours
BASE		NUMBER(5,2)	

5.2.1.39 RAIN

Rainfall data corresponding to specified gauges, times and durations.

Column name	Nulls ?	Type	Remarks
GAUGE	NOT NULL	NUMBER(6)	Rain gauge number using Met. Office convention
TIME	NOT NULL	DATE	Start time (GMT) of rainfall block
DURATION	NOT NULL	NUMBER(5,2)	Duration in hours of rainfall block
RAIN	NOT NULL	NUMBER(4,1)	Total rainfall depth in mm falling in specified duration

Index: RA_GTD on columns GAUGE, TIME, DURATION

Used in views: EVENT_CAAPIn, EVENT_DRAIN, EVENT_HRAIN, EVENT_RAIN

5.2.1.40 RAIN GAUGE CATALOGUE COMMENT

Comments on recording raingauges.

Column name	Nulls ?	Type	Remarks
GAUGE	NOT NULL	NUMBER(6)	Raingauge number using Met. Office convention
L_NUMBER	NOT NULL	NUMBER(2)	Line number of comment on raingauge
TEXT	NOT NULL	CHAR(80)	Comment on raingauge

Index: RGCC_GL on columns GAUGE, L_NUMBER

5.2.1.41 RAIN GAUGE DETAIL

Table of details of rain gauges.

Column name	Nulls ?	Type	Remarks
GAUGE	NOT NULL	NUMBER(6)	Raingauge number using Met. Office convention
NAME		CHAR(56)	Raingauge name
START_YEAR		NUMBER(5)	First year of gauge operation
END_YEAR		NUMBER(5)	Last year of gauge operation
FREQUENCY		CHAR(9)	Frequency of observation
RECORDER		CHAR(6)	Type of recording rain gauge
START_DATE		CHAR(10)	Start date for recording rain gauge
END_DATE		CHAR(10)	End date for recording rain gauge
MEDIUM		CHAR(4)	Recording medium
TELEMETRY		CHAR(4)	Type of telemetry
TYPE		CHAR(2)	Raingauge type
EVAP_PERC		CHAR(7)	Collection of evaporation and

			percolation data
ROUTE		CHAR(2)	Route for transfer of data to Met. Office
OPERATOR		CHAR(40)	Gauge operator
CLIMATE_ST_NO		NUMBER(5)	Climate station number
DATA_BASE_YEAR		NUMBER(5)	First year of computerised rainfall data

Index: RGD_IND on column GAUGE

5.2.1.42 RAIN GAUGE SITE

Table of sites of rain gauges.

Column name	Nulls ?	Type	Remarks
GAUGE	NOT NULL	NUMBER(6)	Raingauge number using Met. Office convention
GRID		NUMBER(3)	Grid system for easting and northing
EAST		NUMBER(5)	National grid reference easting
NORTH		NUMBER(5)	National grid reference northing
ALTITUDE		NUMBER(5)	Altitude of raingauge in metres
SAAR		NUMBER(5)	Standard annual average rainfall
SOURCE		NUMBER(2)	Code to the source of the data items

Index: RGS_IND on column GAUGE
       RGL_EN on columns NORTH, EAST, GAUGE
       RGS_GEN on columns GAUGE, EAST, NORTH

Used in views: RAINGAUGE_LIST

5.2.1.43 RAIN GAUGE SITE SOURCE

Table of sources of raingauge information.

Column name	Nulls ?	Type	Remarks
SOURCE	NOT NULL	NUMBER(2)	Code to source of raingauge information
DESCRIPTION	NOT NULL	CHAR(40)	Description of source of raingauge information

5.2.1.44 RB CATCHMENT COMMENT

Comments relating to catchment areas of representative basins.

Column name	Nulls ?	Type	Remarks
CATCHMENT	NOT NULL	NUMBER(6)	Catchment number using Surface Water Archive convention
SEQUENCE	NOT NULL	NUMBER(2)	Sequence number of text referring to catchment
TEXT	NOT NULL	CHAR(60)	Comments on catchment

Index: RB_CC on columns CATCHMENT, SEQUENCE

5.2.1.45 RB STATION COMMENT

Comments relating to station areas of representative basins.

Column name	Nulls ?	Type	Remarks
CATCHMENT	NOT NULL	NUMBER(6)	Station number using Surface Water Archive convention
SEQUENCE	NOT NULL	NUMBER(2)	Sequence number of text referring to station
TEXT	NOT NULL	CHAR(60)	Comments on station

Index: RB_SC on columns CATCHMENT, SEQUENCE

5.2.1.46 *REFERENCE LIST*

Comments relating to references used.

Column name	Nulls ?	Type	Remarks
TITLE	NOT NULL	CHAR(80)	Title of reference
AUTHOR	NOT NULL	CHAR(80)	Authors of reference
PUBLICATION	NOT NULL	CHAR(80)	Publication in which reference appeared
YEAR	NOT NULL	NUMBER(4)	Year of publication
MONTH		NUMBER(2)	Month of publication
VOLUME		NUMBER(3)	Volume number of publication in which reference appeared
PART		CHAR(4)	Part number of publication in which reference appeared
PAGES	NOT NULL	CHAR(9)	Page numbers of publication in which reference appeared
KEYWORDS	NOT NULL	CHAR(80)	Keywords in reference
<ENTER>ED_BY		CHAR(4)	Initials of person entering reference on table

5.2.1.47 *SCHEME BASEFLOW*

Column name	Nulls ?	Type	Remarks
EVENT_DATE		DATE	
CATCHMENT		NUMBER(6)	
PEAKTIME		DATE	
ENDTIME		DATE	

5.2.1.48 *SCHEME CATCHMENT*

Sub-catchment data for SCHEME.

Column name	Nulls ?	Type	Remarks
CATCHMENT	NOT NULL	NUMBER(6)	
REF	NOT NULL	NUMBER(7,3)	
SEQUENCE	NOT NULL	NUMBER(3)	
FLAG1	NOT NULL	NUMBER(1)	
FLAG2	NOT NULL	NUMBER(1)	
FLAG3	NOT NULL	NUMBER(1)	
EAST		NUMBER(4)	
NORTH		NUMBER(4)	
CHANNEL_LENGTH		NUMBER(6,2)	
CHANNEL_SLOPE		NUMBER(6,2)	
CHANNEL_BREDTH		NUMBER(6,2)	
ROUGHNESS		NUMBER(6,4)	
AREA		NUMBER(5,2)	
LENGTH		NUMBER(6,2)	
SLOPE		NUMBER(6,2)	
IMPERVIOUS_FRACTION		NUMBER(3,2)	
URBAN_FRACTION		NUMBER(3,2)	
GAUGE		NUMBER(6)	

5.2.1.49 *SEARCH POLY*

Polygon definitions of catchment boundaries for searches.

Column name	Nulls ?	Type	Remarks
PID	NOT NULL	NUMBER(6)	Polygon reference number (normally catchment number)
MINY	NOT NULL	NUMBER(5)	Minimum y-value for line
MAXY	NOT NULL	NUMBER(5)	Maximum y-value for line
SLOPE	NOT NULL	NUMBER(12,5)	1/slope of line
CONSTANT	NOT NULL	NUMBER(8,2)	Constant for line or x when slope is zero
FACTOR	NOT NULL	NUMBER(3,1)	Expansion factor used to fill SEARCH_POLY from quadxl,quadyl,etc

Index: SP_PFSC on columns PID, FACTOR, SLOPE, CONSTANT

5.2.1.50 SEARCH_RAIN_SQUARE_DATE

10km squares and dates with data transferred from Cache-Cache to ORACLE.

Column name	Nulls ?	Type	Remarks
EAST	NOT NULL	NUMBER(5)	Easting of south-western corner of grid square
NORTH	NOT NULL	NUMBER(5)	Northing of south-western corner of grid square
DAY	NOT NULL	DATE	Date of data on ORACLE for this square
LOADED	NOT NULL	DATE	Date of transfer of data from Cache-Cache to ORACLE for this square and date
ZONE		NUMBER(3)	Zone code for grid square : -1 for Northern Ireland, 30 for Channel Islands, 0 for rest of UK

Index: SRSD_ALL on columns EAST, NORTH, DAY

Used in views: RAINGAUGE_LIST

5.2.1.51 WRAP

Table of WRAP (Winter Rainfall Acceptance Potential) multipliers. WRAP is the soil classification used in the Flood Studies Report.

Column name	Nulls ?	Type	Remarks
SCHEME	NOT NULL	CHAR(6)	Scheme for which particular WRAP classification is used
CLASS	NOT NULL	NUMBER(1)	Soil class
WEIGHT	NOT NULL	NUMBER(3)	Multiplier for particular scheme and soil class

5.2.2 SWA

5.2.2.1 CATCHMENT_CLIMATE

Column name	Nulls ?	Type	Remarks
CATCHMENT	NOT NULL	NUMBER(6)	
SAAR_4170		NUMBER(5)	
SAAR_1650		NUMBER(5)	
R_JEN		NUMBER(3,2)	
RSMD		NUMBER(4,1)	
EMP2		NUMBER(4,1)	
EMP24		NUMBER(4,1)	
M5_2D		NUMBER(4,1)	
SMDBAR		NUMBER(3,1)	
AGCLIM		NUMBER(3)	

5.2.2.2 CATCHMENT_COMMENT

COMMENTS ON CATCHMENTS FOR PUBLICATION IN HYDROLOGICAL DATA UK 1981-85 STATISTICS VOLUME

Column name	Nulls ?	Type	Remarks
STATION	NOT NULL	NUMBER(6)	
SEQ_NO	NOT NULL	NUMBER(3)	
CAT_COMMENT		CHAR(60)	

5.2.2.3 CATCHMENT_DRAINAGE

Column name	Nulls ?	Type	Remarks
CATCHMENT	NOT NULL	NUMBER(6)	
MSL		NUMBER(6,2)	

DVF		NUMBER(4,2)	
SL1085		NUMBER(7,3)	
STMFRQ		NUMBER(6,3)	
SLTAY		NUMBER(7,3)	
HTSHED		NUMBER(5,1)	
HTMAX		NUMBER(5,1)	
MSL_TRUE		NUMBER(6,2)	

5.2.2.4 CATCHMENT_LANDFORM

Catchment and climate characteristics

Column name	Nulls ?	Type	Remarks
CATCHMENT	NOT NULL	NUMBER(6)	
URBAN		NUMBER(7,3)	Urban fraction
LAKE		NUMBER(7,3)	Fraction of catchment draining through significant lakes
SLOL		NUMBER(7,3)	Overland slope
SHAPK		NUMBER(4,2)	Shape factor k
SOIL1		NUMBER(6,2)	Fraction of WRAP 1 soil on catchment
SOIL2		NUMBER(6,2)	
SOIL3		NUMBER(6,2)	
SOIL4		NUMBER(6,2)	
SOIL5		NUMBER(6,2)	

5.2.2.5 CATCHMENT_RAINFALL_1916_50

1916_50 monthly catchment average rainfall in mm

Column name	Nulls ?	Type	Remarks
CATCHMENT	NOT NULL	NUMBER(6)	
JAN		NUMBER(3)	
FEB		NUMBER(3)	
MAR		NUMBER(3)	
APR		NUMBER(3)	
MAY		NUMBER(3)	
JUN		NUMBER(3)	
JUL		NUMBER(3)	
AUG		NUMBER(3)	
SEP		NUMBER(3)	
OCT		NUMBER(3)	
NOV		NUMBER(3)	
DEC		NUMBER(3)	

5.2.2.6 CATCHMENT_RAINFALL_1941_70

1941_70 monthly catchment average rainfall in mm

Column name	Nulls ?	Type	Remarks
CATCHMENT	NOT NULL	NUMBER(6)	
JAN		NUMBER(3)	
FEB		NUMBER(3)	
MAR		NUMBER(3)	
APR		NUMBER(3)	
MAY		NUMBER(3)	
JUN		NUMBER(3)	
JUL		NUMBER(3)	
AUG		NUMBER(3)	
SEP		NUMBER(3)	
OCT		NUMBER(3)	
NOV		NUMBER(3)	
DEC		NUMBER(3)	

5.2.2.7 DERIVED_FLOW_STATISTIC

Period of record flow statistics for gauging stations

Column name	Nulls ?	Type	Remarks

GAUGE	NOT NULL	NUMBER(6)	
MAF_ANMAX		NUMBER(6,2)	
MAF_BEST		NUMBER(6,2)	
Q10REC		NUMBER(6,2)	
NMF		NUMBER(7,2)	

5.2.2.8 *FACTORS AFFECTING FLOW REGIME*

Factors operating within a catchment which alter natural flow

Column name	Nulls ?	Type	Remarks
GAUGE	NOT NULL	NUMBER(6)	Flow gauge number using Surface Water Archive convention
S_RES		CHAR(1)	S if affected by storage or impounding reservoir
R_AUGMENT		CHAR(1)	R if augmented from surface water and/or groundwater
P_PWS		CHAR(1)	P if affected by abstractions for public water supply
G_GW		CHAR(1)	G if affected by groundwater abstraction and/or recharge
E_EFFLUENT		CHAR(1)	E if augmented by effluent returns
I_IND_AGRI		CHAR(1)	I if reduced by industrial and or agricultural abstractions
H_HEP		CHAR(1)	H if regulated for HEP
N_NATURAL		CHAR(1)	N if natural to within 10% at the 95 percentile flow

5.2.2.9 *HMC DETERMINANDS*

Provides details of each determinand

Column name	Nulls ?	Type	Remarks
CODE	NOT NULL	NUMBER(3)	Numerical code for determinand
NAME	NOT NULL	CHAR(30)	
UNITS		CHAR(15)	
TYPE_SET_UNITS		CHAR(40)	Contains super and sub script commands i.e ¬ and ¬¬

5.2.2.10 *POR FLOW STATISTIC*

Period of record flow statistics for gauging stations

Column name	Nulls ?	Type	Remarks
GAUGE	NOT NULL	NUMBER(6)	
ADF		NUMBER(7,3)	
BFI		NUMBER(3,2)	
UBFI		NUMBER(3,2)	
Q_MAX		NUMBER(7,2)	
YEAR_MAX		DATE	

5.2.2.11 *PREVIOUS NUMBERS*

Former numbers of gauging stations that have been renumbered

Column name	Nulls ?	Type	Remarks
GAUGE	NOT NULL	NUMBER(6)	Current gauge number on UK Surface Water Archive
OLD_NO	NOT NULL	NUMBER(6)	Previous number. (A station may have more than one entry in this table)

5.2.2.12 REGIONAL MONTHLY AREAL RAIN

AREAL MONTHLY RAINFALL(MM) FOR COUNTYs, NRAs, PLCs AND RPBs

Column name	Nulls ?	Type	Remarks
REGION	NOT NULL	CHAR(8)	REGION. UP TO FOUR CHARACTER ABREVIATION
YEAR	NOT NULL	NUMBER(4)	
MONTH	NOT NULL	NUMBER(4)	
RAINFALL	NOT NULL	NUMBER(3)	RAINFALL IN MM

5.2.2.13 STATION CATEGORIES

Gauging station categorisation for Surface Water Archive project

Column name	Nulls ?	Type	Remarks
GAUGE	NOT NULL	NUMBER(6)	Flow gauge number using Surface Water Archive convention
CAT_P		NUMBER(1)	Priority category status. 1reserve 2monthly 3daily
CAT_1		NUMBER(1)	Category 1 status. 1 -1, 2 +1, 3 1
CAT_2A		NUMBER(1)	Category 2a status. 1 -2a, 2 +2a, 3 2a
CAT_2B		NUMBER(1)	Category 2b status. 1 -2b, 2 +2b, 3 2b
CAT_2C		NUMBER(1)	Category 2c status. 1 -2c, 2 +2c, 3 2c
CAT_3		NUMBER(1)	Category 3 status. 1 -3, 2 +3, 3 3
CAT_4		NUMBER(1)	Category 4 status. 1 -4, 2 +4, 3 4
CAT_5		NUMBER(1)	Category 3 status. 1 -5, 2 +5, 3 5

5.2.2.14 STATION COMMENT

COMMENTS ON STATIONS FOR PUBLICATION IN HYDROLOGICAL DATA UK 1981-85 STATISTICS VOLUME

Column name	Nulls ?	Type	Remarks
STATION	NOT NULL	NUMBER(6)	
SEQ_NO	NOT NULL	NUMBER(3)	
STN_COMMENT		CHAR(60)	

5.2.2.15 STATION DESCRIPTION

DESCRIPTIONS OF RIVER FLOW GAUGING STATIONS FOR SURFACE WATER ARCHIVE

Column name	Nulls ?	Type	Remarks
STATION	NOT NULL	NUMBER(6)	
SEQ_NO	NOT NULL	NUMBER(3)	
STN_DESC		CHAR(60)	

5.2.2.16 STATION DETAIL

Locational and administrative details of flow measuring gauges

Column name	Nulls ?	Type	Remarks
GAUGE	NOT NULL	NUMBER(6)	Flow gauge number using Surface Water Archive convention
LOCAL_NUM		NUMBER(10)	
AREA		NUMBER(8,2)	
ZONE		NUMBER(3)	
EAST		NUMBER(5)	
NORTH		NUMBER(5)	
OPERATOR		NUMBER(3)	

ADMIN_REGION		NUMBER(2)	
BEGINS		DATE	
ENDS		DATE	
BEGINS_ACC		CHAR(2)	Accuracy of date in begins. MI, HH, DD or MM (null for year)
ENDS_ACC		CHAR(2)	Accuracy of date in ends. MI, HH, DD or MM (null for year)
ALTITUDE		NUMBER(5,1)	

5.2.2.17 STATION DETAIL TECHNICAL

Technical details of flow measuring stations

Column name	Nulls ?	Type	Remarks
GAUGE	NOT NULL	NUMBER(6)	
BFSTAGE		NUMBER(5,2)	Bank-full stage (m)
BFFLOW		NUMBER(6,2)	Bank-full flow (cumec)
SFSTAGE		NUMBER(4,2)	Structure-full stage (m)
SFFLOW		NUMBER(6,2)	Structure-full flow (cumec)
STYPE		CHAR(8)	Abbreviation for station type. See table station_type_dictionary
FLOOD_WARN		CHAR(1)	Y if a flood warning station. Otherwise null.
PHONE		CHAR(11)	Telemetry telephone number with dialling code.

5.2.2.18 STATION FLOW STATS A

FLOW STATISTICS FOR UK RIVER GAUGING STATIONS (1961-1985)

Column name	Nulls ?	Type	Remarks
GAUGE	NOT NULL	NUMBER(6)	FLOW GAUGE NUMBER USING SURFACE WATER ARCHIVE CONVENTION
MONTH	NOT NULL	NUMBER(2)	
Q_0D5		NUMBER(7,3)	0.5 PERCENTILE FLOW
Q_99D5		NUMBER(7,3)	99.5 PERCENTILE FLOW
Q_AVE		NUMBER(7,3)	AVERAGE DAILY MEAN FLOW
Q_MIN		NUMBER(7,3)	MINIMUM DAILY MEAN FLOW
Q_MAX		NUMBER(7,3)	MAXIMUM DAILY MEAN FLOW
RISE_5TH_ABS		NUMBER(7,3)	FIFTH HIGHEST ABSOLUTE RISE IN DISCHARGE
FALL_5TH_PC		NUMBER(6,3)	FIFTH HIGHEST PERCENT FALL IN DISCHARGE
STATS_START		NUMBER(4)	FIRST YEAR OF DATA USED TO DERIVE STATS
STATS_END		NUMBER(4)	LAST YEAR OF DATA USED TO DERIVE STATISTICS
STATS_YEARS		NUMBER(3)	NUMBER OF YEARS OF RECORD USED TO DERIVE STATS BETWEEN GIVEN RANGE

5.2.2.19 STATION FLOW STATS B

Period-of-record river flow statistics. Continually updated

Column name	Nulls ?	Type	Remarks
GAUGE	NOT NULL	NUMBER(6)	River gauge number on UK Surface Water Archive
Q_BAR		NUMBER(7,3)	Mean gauged flow (cumec) from all years with <6 missing days
Q_INST_MAX		NUMBER(7,2)	Highest instantaneous maximum flow (from SWA HIFLOWS data)
Q_I_M_DATE		DATE	Date of Q_INST_MAX
Q_D_ACC		CHAR(2)	Accuracy of Q_I_M_DATE. Nullday MMmonth YYyear
Q_BEGIN		NUMBER(4)	First year used for Q_BAR
Q_END		NUMBER(4)	Last year used for Q_BAR
Q_YEARS		NUMBER(3)	Number of years used for Q_BAR
Q_UPDATE		DATE	Date when Q_BAR or Q_INST_MAX were last updated
MAF		NUMBER(7,2)	Mean annual flood derived from POT

			database
AF_MAX		NUMBER(7,2)	Highest flood from POT database
AF_M_DATE		DATE	Date of AF_MAX
AF_D_ACC		CHAR(2)	Accuracy of AF_M_DATE. Nullday MMmonth YYyear
AF_BEGIN		NUMBER(4)	First year used for AF statistics
AF_END		NUMBER(4)	Last year used for AF statistics
AF_YEARS		NUMBER(3)	Number of years used for AF statistics
AF_UPDATE		DATE	Date when AF statistics were last updated
BFI		NUMBER(3,2)	Baseflow index (proportion of flow which is baseflow)
BFI_BEGIN		NUMBER(4)	First year used for BFI derivation
BFI_END		NUMBER(4)	Last year used for BFI derivation
BFI_YEARS		NUMBER(3)	Number of years used for BFI
BFI_UPDATE		DATE	Date when BFI was last updated

5.2.2.20 STATION NAME

Names of flow measuring gauges

Column name	Nulls ?	Type	Remarks
GAUGE	NOT NULL	NUMBER(6)	Flow gauge number using Surface Water Archive convention
RIVER	NOT NULL	CHAR(32)	Name of river
LOCATION	NOT NULL	CHAR(38)	Name of location on river

5.2.2.21 STATION OPERATOR DICTIONARY

Column name	Nulls ?	Type	Remarks
OPERATOR	NOT NULL	NUMBER(3)	
ABBREVIATION		CHAR(8)	
FULL_NAME		CHAR(100)	
ADDRESS		CHAR(100)	
POSTCODE		CHAR(100)	
TELEPHONE		CHAR(11)	
YEAR_CLOSED		NUMBER(4)	

5.2.2.22 STATION TYPE DICTIONARY

Expansion of the station type code used in other SWA tables

Column name	Nulls ?	Type	Remarks
STYPE	NOT NULL	NUMBER(3)	
ABBREVIATION		CHAR(8)	
DESCRIPTION		CHAR(240)	

5.2.3 Others

5.2.3.1 NJB.CHARACTERISTICS

Automatically derived catchment characteristics.

Column name	Nulls ?	Type	Remarks
CATCHMENT	NOT NULL	NUMBER(6)	Catchment number (those on dig.bounduk_hdr)
LAST_UPDATE		CHAR(8)	Date of last change to row
AREA		NUMBER(8,2)	Area on dig.bounduk_hdr (sq.km)
JUNCTIONS		NUMBER(6)	not set
STMFRQ		NUMBER(8,2)	not set
MSL		NUMBER(8,2)	not set
S1085		NUMBER(8,2)	not set
SOIL1		NUMBER(5,3)	Percentage of scanned area occupied by 100m WRAP class 1
SOIL2		NUMBER(5,3)	Percentage of scanned area occupied

		by 100m WRAP class 2
SOIL3	NUMBER(5,3)	Percentage of scanned area occupied by 100m WRAP class 3
SOIL4	NUMBER(5,3)	Percentage of scanned area occupied by 100m WRAP class 4
SOIL5	NUMBER(5,3)	Percentage of scanned area occupied by 100m WRAP class 5
SOIL	NUMBER(4,3)	0.15soil1+0.30soil2+0.40soil3+0.45soil4+0.50soil5; with soils being expressed as fractions, not percentages
SAAR	NUMBER(4)	Standard (1941-70) Annual Average Rainfall; 1km grid with interpolation
MIN_SAAR	NUMBER(4)	
MAX_SAAR	NUMBER(4)	
T2_DAY_M5	NUMBER(3)	Two day rainfall (mm) with return period of five years; 1km grid with interpolation
MIN_T2_DAY_M5	NUMBER(3)	
MAX_T2_DAY_M5	NUMBER(3)	
R	NUMBER(3)	not set
MIN_R	NUMBER(3)	
MAX_R	NUMBER(3)	
SMD	NUMBER(3,1)	not set
MIN_SMD	NUMBER(3,1)	
MAX_SMD	NUMBER(3,1)	
RSMD	NUMBER(6,2)	not set
LAKE	NUMBER(5,2)	not set
URBAN	NUMBER(5,2)	not set
EMR_2HR	NUMBER(3)	not set
MIN_EMR_2HR	NUMBER(3)	
MAX_EMR_2HR	NUMBER(3)	
EMR_24HR	NUMBER(3)	not set
MIN_EMR_24HR	NUMBER(3)	
MAX_EMR_24HR	NUMBER(3)	
T2_DAY_M5_PC_AAR	NUMBER(3,1)	not set
MIN_T2_DAY_M5_PC_AAR	NUMBER(3,1)	
MAX_T2_DAY_M5_PC_AAR	NUMBER(3,1)	
T25_DAY_M5_PC_AAR	NUMBER(2)	not set
MIN_T25_DAY_M5_PC_AAR	NUMBER(3,1)	
MAX_T25_DAY_M5_PC_AAR	NUMBER(3,1)	
PE	NUMBER(3)	From mapped 1km PE grid
MIN_PE	NUMBER(3)	
MAX_PE	NUMBER(3)	
EP	NUMBER(5)	Effective precipitation from MORECS data - annual average 1961-83 in x 10 mm units
MIN_EP	NUMBER(5)	
MAX_EP	NUMBER(5)	

5.3 *Views*

5.3.1 Flood

All the views with names 'EVENT_data type' are joins of the tables holding the time series data with the corresponding entries in the COMPONENT table. Thus a select from an 'EVENT_data type' view specifying the event number will return all the data of that type for all of the sites having that data type. For example `select * from EVENT_FLOW where event=n;` will return all flows for the site between the start and end times in the COMPONENT table. All the views have exactly the same columns so that they can be unioned to return many data types with one select. This is why the time series data are all found in a column called DATA_VALUE. For data types with no interval (eg. flows which are instantaneous) the intervals are set to -1. The column ELAPSED_TIME makes it very easy for a FORTRAN program to calculate data interval which is useful because ORACLE cannot (easily) perform the interval checking. If half-hourly data are contained in the FLOW table, but the COMPONENT asks for hourly data, the view will contain all the half hourly values, but in practice this has not proved a problem.

The following sections give details of each view, with descriptions of the columns, an indication of those columns that are filled, and notes on tables from which the view is created.

5.3.1.1 *EVENT CAAPI1*

Join of tables: COMPONENT, EVENT, RAIN.

Column name	Nulls ?	Type	Remarks
EVENT	NOT NULL	NUMBER(4)	Event number on ORACLE
DATA_TYPE	NOT NULL	NUMBER(2)	Data type 6 for Autostorm
SITE	NOT NULL	NUMBER(6)	Catchment number
DATE_TIME	NOT NULL	CHAR(75)	Date/time of value
ELAPSED_TIME	NOT NULL	NUMBER	Time since 1 January 1980 in seconds
DATA_VALUE	NOT NULL	NUMBER(4,1)	Autostorm catchment average API value
INTERVAL	NOT NULL	NUMBER(5,2)	Block duration for data in hours

5.3.1.2 *EVENT CAAPI2*

Join of tables: COMPONENT, EVENT, RAIN.

Column name	Nulls ?	Type	Remarks
EVENT	NOT NULL	NUMBER(4)	Event number on ORACLE
DATA_TYPE	NOT NULL	NUMBER(2)	Data type 7 for SCRAP
SITE	NOT NULL	NUMBER(6)	Catchment number
DATE_TIME	NOT NULL	CHAR(75)	Date/time of value
ELAPSED_TIME	NOT NULL	NUMBER	Time since 1 January 1980 in seconds
DATA_VALUE	NOT NULL	NUMBER(4,1)	SCRAP catchment average API value
INTERVAL	NOT NULL	NUMBER(5,2)	Block duration for data in hours

5.3.1.3 *EVENT CAAPI3*

Join of tables: COMPONENT, EVENT, RAIN.

Column name	Nulls ?	Type	Remarks
EVENT	NOT NULL	NUMBER(4)	Event number on ORACLE
DATA_TYPE	NOT NULL	NUMBER(2)	Data type 17 for CARP
SITE	NOT NULL	NUMBER(6)	Catchment number
DATE_TIME	NOT NULL	CHAR(75)	Date/time of value
ELAPSED_TIME	NOT NULL	NUMBER	Time since 1 January 1980 in seconds
DATA_VALUE	NOT NULL	NUMBER(4,1)	CARP catchment average API value
INTERVAL	NOT NULL	NUMBER(5,2)	Block duration for data in hours

5.3.1.4 *EVENT CARAIN1*

Join of tables: COMPONENT, EVENT, CARAIN1.

Column name	Nulls ?	Type	Remarks
EVENT	NOT NULL	NUMBER(4)	Event number on ORACLE
DATA_TYPE	NOT NULL	NUMBER(2)	Data type 2 for Autostorm
SITE	NOT NULL	NUMBER(6)	Catchment number
DATE_TIME	NOT NULL	CHAR(75)	Date/time of value
ELAPSED_TIME	NOT NULL	NUMBER	Time since 1 January 1980 in seconds
DATA_VALUE	NOT NULL	NUMBER(4,1)	Autostorm catchment average rainfall
INTERVAL	NOT NULL	NUMBER(5,2)	Block duration for data in hours

5.3.1.5 *EVENT CARAIN2*

Join of tables: COMPONENT, EVENT, CARAIN2.

Column name	Nulls ?	Type	Remarks
EVENT	NOT NULL	NUMBER(4)	Event number on ORACLE

DATA_TYPE	NOT NULL	NUMBER(2)	Data type 3 for SCRAP
SITE	NOT NULL	NUMBER(6)	Catchment number
DATE_TIME	NOT NULL	CHAR(75)	Date/time of value
ELAPSED_TIME	NOT NULL	NUMBER	Time since 1 January 1980 in seconds
DATA_VALUE	NOT NULL	NUMBER(4,1)	SCRAP catchment average rainfall
INTERVAL	NOT NULL	NUMBER(5,2)	Block duration for data in hours

5.3.1.6 EVENT_CARAIN3

Join of tables: COMPONENT, EVENT, CARAIN3.

Column name	Nulls ?	Type	Remarks
EVENT	NOT NULL	NUMBER(4)	Event number on ORACLE
DATA_TYPE	NOT NULL	NUMBER(2)	Data type 13 for CARP
SITE	NOT NULL	NUMBER(6)	Catchment number
DATE_TIME	NOT NULL	CHAR(75)	Date/time of value
ELAPSED_TIME	NOT NULL	NUMBER	Time since 1 January 1980 in seconds
DATA_VALUE	NOT NULL	NUMBER(4,1)	CARP catchment average rainfall
INTERVAL	NOT NULL	NUMBER(5,2)	Block duration for data in hours

5.3.1.7 EVENT_CARAIN4

Join of tables: COMPONENT, EVENT, CARAIN4.

Column name	Nulls ?	Type	Remarks
EVENT	NOT NULL	NUMBER(4)	Event number on ORACLE
DATA_TYPE	NOT NULL	NUMBER(2)	Data type 99 for Radar
SITE	NOT NULL	NUMBER(6)	Catchment number
DATE_TIME	NOT NULL	CHAR(75)	Date/time of value
ELAPSED_TIME	NOT NULL	NUMBER	Time since 1 January 1980 in seconds
DATA_VALUE	NOT NULL	NUMBER(4,1)	Radar catchment average rainfall
INTERVAL	NOT NULL	NUMBER(5,2)	Block duration for data in hours

5.3.1.8 EVENT_CASMD

Join of tables: COMPONENT, EVENT, CASMD.

Column name	Nulls ?	Type	Remarks
EVENT	NOT NULL	NUMBER(4)	Event number on ORACLE
DATA_TYPE	NOT NULL	NUMBER(2)	Data type 4 for catchment SMD
SITE	NOT NULL	NUMBER(6)	Catchment number
DATE_TIME	NOT NULL	CHAR(75)	Date/time of value
ELAPSED_TIME	NOT NULL	NUMBER	Time since 1 January 1980 in seconds
DATA_VALUE	NOT NULL	NUMBER(4,1)	Catchment average SMD
INTERVAL	NOT NULL	NUMBER(5,2)	Data interval in hours

5.3.1.9 EVENT_DRAIN

Join of tables: COMPONENT, EVENT, RAIN (new version of view DRAIN renamed August 1990).

Column name	Nulls ?	Type	Remarks
EVENT	NOT NULL	NUMBER(4)	Event number on ORACLE
SITE	NOT NULL	NUMBER(6)	Raingauge site
TIME	NOT NULL	DATE	Date/time of value
RAIN	NOT NULL	NUMBER(4,1)	Daily rainfall

5.3.1.10 EVENT_FLOW

Join of tables: COMPONENT, EVENT, FLOW.

Column name	Nulls ?	Type	Remarks

EVENT	NOT NULL	NUMBER(4)	Event number on ORACLE
DATA_TYPE	NOT NULL	NUMBER(2)	Data type 1 for flow
SITE	NOT NULL	NUMBER(6)	Flow gauge number
DATE_TIME	NOT NULL	CHAR(75)	Date/time of value
ELAPSED_TIME	NOT NULL	NUMBER	Time since 1 January 1980 in seconds
DATA_VALUE	NOT NULL	NUMBER(4,1)	Flow
INTERVAL	NOT NULL	NUMBER(5,2)	Data interval in hours (set to -1)

5.3.1.11 *EVENT_HRAIN*

Join of tables: COMPONENT, EVENT, RAIN.

Column name	Nulls ?	Type	Remarks
EVENT	NOT NULL	NUMBER(4)	Event number on ORACLE
DATA_TYPE	NOT NULL	NUMBER(2)	Data type 10 for rainfall
SITE	NOT NULL	NUMBER(6)	Raingauge site
TIME	NOT NULL	DATE	Date/time of value
DATA_VALUE	NOT NULL	NUMBER(4,1)	Hourly rainfall

5.3.1.12 *EVENT_RAIN*

Join of tables: COMPONENT, EVENT, RAIN.

Column name	Nulls ?	Type	Remarks
EVENT	NOT NULL	NUMBER(4)	Event number on ORACLE
DATA_TYPE	NOT NULL	NUMBER(2)	Data type 10 for rainfall
SITE	NOT NULL	NUMBER(6)	Raingauge site
DATE_TIME	NOT NULL	CHAR(75)	Date/time of value
ELAPSED_TIME	NOT NULL	NUMBER	Time since 1 January 1980 in seconds
DATA_VALUE	NOT NULL	NUMBER(4,1)	Rainfall
INTERVAL	NOT NULL	NUMBER(5,2)	Block duration for data in hours

5.3.1.13 *FLOOD_ESTIMATE_ANMAX_DETAILS*

Join of tables: FLOOD_ESTIMATE_ANMAX_ORIGINAL, FESTFITS.

Column name	Nulls ?	Type	Remarks
CATCHMENT	NOT NULL	NUMBER(6)	
FIRST_YEAR		NUMBER(4)	
LAST_YEAR		NUMBER(4)	
YEARS_OF_DATA		NUMBER(3)	
FIT_METHOD		CHAR(3)	
MAX_RETURN_PERIOD		NUMBER(3)	
QUALITY_EXTRAP		NUMBER(2)	
GEV_U		NUMBER(6,2)	
GEV_A		NUMBER(6,2)	
GEV_K		NUMBER(6,2)	
MAD		NUMBER(5,3)	
TMAX100		NUMBER(5,1)	
TMAX125		NUMBER(5,1)	
TMAX150		NUMBER(5,1)	

5.3.1.14 *FLOOD_ESTIMATE_BY_FREQUENCY*

From table FLOOD_ESTIMATE.

Column name	Nulls ?	Type	Remarks
CATCHMENT	NOT NULL	NUMBER(6)	Catchment number
METHOD	NOT NULL	NUMBER(1)	Method of flood estimates
			Flood estimates by that method for various return periods
T2		NUMBER	
T5		NUMBER	
T10		NUMBER	
T25		NUMBER	

T50		NUMBER	
T100		NUMBER	

5.3.1.15 FLOOD ESTIMATE BY METHOD

From table FLOOD_ESTIMATE.

Column name	Nulls ?	Type	Remarks
CATCHMENT	NOT NULL	NUMBER(6)	Catchment number
RETURN PERIOD	NOT NULL	NUMBER(4)	Return period of flood estimate in years
			Flood estimates at that return period for various methods
DATA_STATS		NUMBER	
FSR_STATS		NUMBER	
FSR_UH		NUMBER	
SR16_UH		NUMBER	
SR16_LTPPR		NUMBER	
SR16_LTP		NUMBER	
SR16_LPR		NUMBER	
SR16_URB		NUMBER	
SR16_URB_L		NUMBER	

5.3.1.16 FLOWRAIN

Column name	Nulls ?	Type	Remarks
SITE		NUMBER	
BEGIN		DATE	
TIME		DATE	
FLOW		NUMBER	
RAIN		NUMBER	

5.3.1.17 NSHB RESULT

Join of tables: COMPONENT, FSSR16_RESULT.

Column name	Nulls ?	Type	Remarks
CATCHMENT	NOT NULL	NUMBER	Catchment number
BEGIN	NOT NULL	DATE	Start date-time of required data
RTOTAL		NUMBER	Rainfall depth in mm
FLPEAK		NUMBER	Flow peak in cumec
ANTECF		NUMBER	Flow at start of hydrograph rise in cumec
PR		NUMBER	Percentage runoff
SMD		NUMBER	Soil moisture deficit at start of rainfall in mm
API		NUMBER	Antecedent precipitation index at start of rainfall in mm
TPEAK		NUMBER	Unit hydrograph time to peak in hours
QPEAK		NUMBER	Unit hydrograph peak in cumec/100 sq.km

5.3.1.18 RAINGAUGE LIST

Join of tables: RAIN_GAUGE_SITE, SEARCH_POLY.

This is a smart view that uses a point-in-polygon routine supplied by NCS (Robert Sanderson). It makes a search for raingauges entered in table RAIN_GAUGE_SITE that are within the quad defined by a given PID (renamed CATCHMENT here) and FACTOR in table SEARCH_POLY.

Column name	Nulls ?	Type	Remarks
CATCHMENT	NOT NULL	NUMBER(6)	Catchment number (polygon reference number)
FACTOR	NOT NULL	NUMBER(3,1)	Expansion factor used to fill SEARCH_POLY from quadx1,quady1,etc
GAUGE	NOT NULL	NUMBER(6)	Raingauge number using Met. Office

			convention
EAST		NUMBER(5)	National grid reference easting
NORTH		NUMBER(5)	National grid reference northing

5.3.1.19 UH ANALYSIS RESULT

Join of tables: COMPONENT, FSSR16_RESULT.

Column name	Nulls ?	Type	Remarks
CATCHMENT	NOT NULL	NUMBER	Catchment number
START_DATE	NOT NULL	DATE	Start date-time of required data
RTOTAL		NUMBER	Rainfall depth in mm
FLPEAK		NUMBER	Flow peak in cumec
ANTECF		NUMBER	Flow at start of hydrograph rise in cumec
PR		NUMBER	Percentage runoff
SMD		NUMBER	Soil moisture deficit at start of rainfall in mm
API		NUMBER	Antecedent precipitation index at start of rainfall in mm
TPEAK		NUMBER	Unit hydrograph time to peak in hours
QPEAK		NUMBER	Unit hydrograph peak in cumec/100 sq.km
WHALEP		NUMBER	Unit hydrograph width at half the peak value in hours
INTERVAL	NOT NULL	NUMBER	Data interval or block duration for data in hours

5.3.2 SWA

5.3.3 Others

6.0 References

Acreman, M.C. (1986) On estimating a catchment average rainfall total, Applied Hydrology Informal Note No. 100.

Jones, S.B. (1983) The estimation of catchment average point rainfall profiles, Institute of Hydrology Report No. 87.

Kidd, C.H.R. & Packman, J.C. (1980) Selection of design storm and antecedent conditions for urban drainage design, Institute of Hydrology Report No. 61.

Appendix A. Dates of changes between GMT and BST

Clocks go forward in the spring and back in the autumn (spring forward, fall back!).

	To		From	
1956	April	22	Oct	7
1957		14		6
1958		20		5
1959		19		4
1960		10		2
1961	March	26		29
1962		25		28
1963		21		27
1964		22		25
1965		21		24
1966		20		23
1967		19		29
1968	Feb	18		

From 18th February 1968 until 28th October 1972
B.S.T. was in use throughout.

1972			Oct	28
1973	March	17		27
1974		16		26
1975		15		25
1976		20		23
1977		20		23
1978		19		29
1979		18		28
1980		16		26
1981		22		25
1982		28		24
1983		27		23
1984		25		28
1985		31		27
1986		16		26
1987		29		25
1988		27		23
1989		26		29
1990		25		28
1991		31		27