RESEARCH MANAGEMENT INFORMATION SYSTEM (REMISS)

Programmer’s Reference Manual - Version 1.00

By Neil R. Sumner

TECHNICAL MEMORANDUM 91/6
March 1991

Division of Water Resources
RESEARCH MANAGEMENT INFORMATION SYSTEM (REMISS)

Programmer's Reference Manual - Version 1.00

By Neil R. Sumner
Division of Water Resources, Perth Laboratory

Technical Memorandum 91/6
March 1991
CSIRO
Institute of Natural Resources and Environment
Division of Water Resources

ISBN 0 643 05133 3
Publications enquiries to:

Divisional Editor
CSIRO Division of Water Resources
GPO Box 1666
Canberra ACT 2601 Australia
ph. (062) 46 5313
Abstract

REsearch Management Information SyStem (REMISS) is a management information system developed by the CSIRO Division of Water Resources to facilitate the management and planning of research. REMISS records and maintains operational and financial information relating to research projects, which is used for the decision-making processes involved in the management of a CSIRO Division. REMISS is also able to supply information to large organisational databases such as the CSIRO Project Database. This document is a reference manual for programmers who are required to maintain the REMISS database; it contains a description of the database and application software.
CONTENTS

1. Introduction

2. Installation
   2.1 System Requirements
   2.2 System Configuration

3. The REMISS Application
   3.1 Application Module
   3.2 Application Support Module
   3.3 Other Procedures
   3.4 Document Files

4. The REMISS Database
   4.1 Database Structure
   4.2 Database Tables
   4.3 Database Columns

5. The REMISS Menu System
   5.1 Main Menu
   5.2 System Description Menu
   5.3 Data Entry Menu
   5.4 Editing Subproject Information Menu
   5.5 Management Reports Menu
   5.6 Financial Reports Menu
   5.7 Print Data Entry forms Menu
   5.8 Database Administration Menu

6. Printers Supported by REMISS
   6.1 Printer Initialisation
   6.2 Special Report Requirements

7. Recovering Unusable Disk Space

References

Appendices
   A. Program Documentation
   B. REMISS Menu Structure
1. INTRODUCTION

REsearch Management Information SyStem (REMISS) (Sumner and Curtis, 1988) is a database for managing research. REMISS has been developed to record the management and financial aspects of research in a CSIRO Division. The database is used to report on the research being undertaken, and to facilitate the allocation of funds.

The database was developed by the former CSIRO Division of Groundwater Research. In this Division research efforts were organized into Projects and Subprojects. Research is carried out at the Subproject level, while Projects are groupings of similar Subprojects.

This programmers reference manual contains the information required by a programmer for the maintenance of REMISS. This manual assumes the reader is familiar with database design and general programming principles. A programmer maintaining REMISS will need access to the REMISS User’s Guide (Sumner, 1991), Rbase User’s Manual (Microsoft, 1985a), and Rbase Reference Manual (Microsoft, 1985b). The manual also assumes the reader has access to the source code for the many programs that together form REMISS.

2. INSTALLATION

2.1 System Requirements

REMISS is designed to run on IBM PC, PC-XT, PC-AT, and compatible microcomputers. The system must have at least 512K of memory, and a hard disk with 2 megabytes of storage available. The system must run MS-DOS version 2.0 or higher. A printer is required if reports are to be produced.

2.2 System Configuration

REMISS requires Rbase run time support in the form of executable files. These files must be stored in a sub-directory from the root directory called "RBASE". The REMISS database and programs are stored in a sub-directory from the RBASE directory called "REMISS".

Users of REMISS only have access to compiled versions of the software. The Rbase and REMISS files required by REMISS users in the respective directories are as follows:

```
C\RBASE
  RBASE.EXE : run time support.
  RBASE.OVL : run time support.
  MESSAGE.RBS : run time support.
  INITIAL.RBS : run time support.
  DEFAULT.RBS : default file.

C\RBASE\REMISS
  REMISS1.RBS : database structure
  REMISS2.RBS : REMISS data.
  REMISS3.RBS : index for key columns.
  RBASE.DAT : start-up file.
  MENU : command file.
  REMISS.APX : application file.
  REMISS2.APX : application support file.
  FORMS1.DOC : data entry forms (ASCII).
```
A programmer responsible for the maintenance of the REMISS system will require the REMISS source programs and the complete Rbase software.

Rbase requires that your system has system configuration file (config.sys) with the following commands:

```plaintext
files=20
buffers=15
```

3. The REMISS Application

The REMISS directory contains several different types of files. Files with the extension ".RBS" form the database while the remaining files are part of the Application. The REMISS Application is the software that provides the user with the menu system and procedures to support the many menu options.

3.1 Application Module

The Application Module contains many compiled command files called binary command blocks. These blocks, more conveniently called procedures can be called by other procedures from inside or external to the same module. This Application Module was created by the Rbase EXPRESS applications generator. The Application Module has the file name "REMISS.APX" and can be found in the REMISS directory.

The following binary command blocks are contained in the REMISS.APX file:

```
REMISS MAIN MENU1 MENU2 MENU3
MENU4 MENU5 MENU6 MENU7 ADHOC
DESC1 DESC2 DESC3 DESC4 DESC5
DESC6 ENTRY1 ENTRY2 EDIT1 EDIT2
EDIT3 EDIT4 EDIT5 MREPORT1
MREPORT2 MREPORT3 MREPORT4 FREPORT1 FREPORT2
FREPORT3 FORMS1 FORMS2 FORMS3 ADMIN1
ADMIN2 ADMIN3 ADMIN4 ADMIN5
```

3.2 Application Support Module

The Application Support Module also contains compiled command files. This module was not created by the Rbase compiler rather than EXPRESS. The file is called "REMISS2.APX" and contains the low level support procedures for the menu driven application.

The following binary command blocks are contained in the REMISS2.APX file:

```
GETYEARS EDIT UNLOAD MREPORT4& ESTENTRY
FREPORT1& LOAD ESTFORMS NOTAVAIL
```

3.3 Other Procedures

Other procedures are also required by the REMISS database. The RBASE.DAT file contains the REMISS initialisation procedure. This procedure checks the date setting on the microcomputers clock. If the setting is correct the REMISS main menu is invoked.
The only other procedure not found in the application modules is the MENU file. This compiled command file is used to return a user to the REMISS main menu from Command Mode.

3.4 Document Files

Document files are ASCII files that are either printed or displayed by REMISS. These files are used by the System Description Menu, and Print Data Entry Forms Menu. An example is the "FORMS1.DOC" file which contains the Initial Forecast Forms. These forms do not contain any information from the database, they are simply printed by REMISS when required.

4. THE REMISS DATABASE

4.1 Database Structure

The REMISS database initially contained 19 tables and 93 different attributes or column names. These numbers are expected to increase as the information requirements of the Division grows. A relational model of the data is used. The database structure can be viewed by the Rbase "LIST ALL" command.

4.2 Database Tables

The attributes contained in a table are determined by the relational model used. On occasions separate tables have been used where one table would normally suffice, because of a constraint imposed by Rbase the maximum number of characters per row is 1530. The tables comprising the database can be viewed by the Rbase "LIST TABLES" command.

4.3 Database Columns

Each table in the database contains two or more columns. Several columns such as "SUBPROJ#" are found in more than one table. These columns are used to index one table with another. The columns are of different types depending on the nature of the data to be stored. The name and type of the database columns can be viewed with the Rbase "LIST COLUMNS" command.

5. THE REMISS MENU SYSTEM

The REMISS menu system was developed with the Rbase EXPRESS applications generator. EXPRESS was used to set-up the menu driven structure of REMISS. The menu options that select another menu or exit from the current menu were implemented from within EXPRESS. The other menu options are external procedures called by the menu program. These compiled procedures or blocks are contained in the binary the application files "REMISS.APX" and "REMISS2.APX". The menu program is in the "REMISS.APX" file. The external procedures are described in Appendix A, and the menu structure is shown as Appendix B.

5.1 Main Menu

Most of the options on the Main Menu simply call another lower level menu. The one exception is the Command Mode option which exits from the REMISS menu program into Rbase. This option should be used to access features of Rbase not available via the menu system. The REMISS user can return to the REMISS main menu by running the "MENU" program. The procedures called by this menu are as follows:
i) Command Mode (ADHOC)

The name of the procedure block is enclosed in parenthesis. The procedure blocks have the same name as the ASCII source file. The ASCII source file however has the extension "CMD". This is an Rbase convention that identifies the file as containing Rbase commands.

5.2 System Description Menu

The System Description Menu does not call external procedures. The menu options simply display an ASCII file containing a description of the selected subject.

5.3 Data Entry Menu

There are two separate data entry procedures which are as follows:

   i) Enter New Subproject Forecast (ENTRY1).
   ii) Enter Annual Estimates (ENTRY2).

5.4 Editing Subproject Information Menu

There are six separate procedures for editing data from the database. These are as follows:

   i) List Subprojects (EDIT1).
   ii) Edit Subproject Descriptive Information (EDIT2).
   iii) Edit Appropriation Finances (EDIT3).
   iv) Edit Contributory Finances (EDIT4).
   v) Edit Appropriation Staff Time Allocation (EDIT5).
   vi) Edit Contributory Staff Time Allocation (EDIT6).

5.5 Management Reports Menu

The standard reports available vary greatly in complexity. The reports are printed by procedures called from the menu program. The Management Reports are as follows:

   i) Inventory (MREPORT1)
   ii) Objectives (MREPORT2)
   iii) Statistics (MREPORT3)
   iv) Descriptive Information (MREPORT4)

5.6 Financial Reports Menu

This menu is similar to the Management Reports Menu. It also allows the user the option of printing certain reports for one Subproject, or for all Subprojects in the database. The Financial Reports available are as follows:

   i) Operating Expenditure Estimates (FREPORT1)
   ii) Operating Expenditure Estimates (FREPORT2)
   iii) Subproject Funding (FREPORT3)
5.7 Print Data Entry Forms Menu

The data entry forms are similar to reports. The only difference is that the forms contain additional blank fields for entering information. The Initial Forecast Forms do not contain any database information and simply consist of an ASCII file which is sent to the printer. The data entry forms available are as follows:

i) Initial Forecast Forms (FORMS1)
ii) Annual Estimates (FORMS2)
iii) Annual Estimates (FORMS3)

5.8 Database Administration Menu

The Database Administration menu provides facilities for updating database tables. The options available are as follows:

i) Update Subproject Status (ADMIN1)
ii) Update Subproject Application Area (ADMIN2)
iii) Update Abbreviated Title (ADMIN3)
iv) Update Project Groups (ADMIN4)
v) Update Staff Classification and Salaries (ADMIN5)

6. PRINTERS SUPPORTED BY REMISS

The REPORTS and FORMS produced by REMISS require an 80 column printer. Any printer capable of printing 80 columns such as the EPSON FX-80, or EPSON FX-100 will suffice. The character pitch on some printers such as the Impact Laser 800 may need to be changed to print 80 columns. One of the standard reports also requires a printer which supports condensed mode printing. REMISS must be installed to support the type of printer available.

6.1 Printer Initialisation

This section describes how to install REMISS to use a printer which requires the character pitch to be changed (Eg Impact Laser 800). These printers require a special initialisation program (IMPACT1.BAS) that is executed prior to using REMISS. Another decommissioning program (IMPACT2.BAS) is also required after using remiss to return the printer to its original state.

These BASIC programs should not be visible to the user. The programs can be easily set-up in a MS-DOS batch file as shown:

```
GWBASIC IMPACT1.BAS
RBASE
GWBASIC IMPACT2.BAS
```

6.2 Special Report Requirements

It is often not possible to fit all the columns required in a report within the 80 column limit. In these cases the condensed mode capabilities of the printer available should be used. This is the case with the Operating Expenditure Estimates financial report.

The Impact Laser 800 printer does not support condensed mode. A similar effect can be achieved by reducing the character pitch. The printer must be returned to its original state after printing these reports. The printer control characters
required should be included in the Rbase report. The ASCII decimal equivalents of the printer control characters for two common printers are shown in parenthesis:

<table>
<thead>
<tr>
<th>PRINTER</th>
<th>SET CONDENSED MODE</th>
<th>RESET PRINTER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Impact Laser 800</td>
<td>{27 67 80 49 54 46}</td>
<td>{27 70 67 80 50 57 46}</td>
</tr>
<tr>
<td>Epson FX-105</td>
<td>(15)</td>
<td>(18)</td>
</tr>
</tbody>
</table>

Impact Laser 800 printer owners should note that the lead in character for this printer has been changed from the factory setting of ASCII 124 to ASCII 27.

7. RECOVERING UNUSABLE DISK SPACE

When rows are deleted from tables in the database the disk space used by these rows is not immediately recovered. The Rbase "PACK" command should be used to reorganize the database and free the disk space. The database must be backed up with the MS-DOS "BACKUP" command first. It is the responsibility of the programmer to ensure the database is using the available disk space efficiently.

REFERENCES


APPENDIX A
PROGRAM DOCUMENTATION

DATA ENTRY PROCEDURES

PROCEDURE NAME : ENTRY1.CMD

FILE NAME : remiss.apx (binary)

PROGRAMMER : Neil Sumner

PURPOSE : Enter initial forecast information for new Subproject.

VARIABLES : spn : Subproject number
            scX#Y : screen X field number Y

DESCRIPTION : Uses data entry screens to accept information for a new Subproject. The information is then saved in the database once data entry is complete.

PROCEDURE NAME : ENTRY2.CMD

FILE NAME : remiss.apx (binary)

PROGRAMMER : Neil Sumner

PURPOSE : Enter annual estimates information.

VARIABLES : spn : Subproject number
            scX#Y : screen X field number Y

DESCRIPTION : This is a driver program that calls other procedures to retrieve data from database, update the data, and then save the data back in the database. Some of the data can be updated, other data must be entered. This procedure can only be used for existing Subprojects.

PROCEDURE NAME : ESTENTRY.CMD

FILE NAME : remiss2.apx (binary)

PROGRAMMER : Neil Sumner

PURPOSE : Entry of annual estimate information into database.

VARIABLES : spn : Subproject number
            scX#Y : screen X field number Y

DESCRIPTION : Procedure to update existing data, and enable entry of new data for annual estimates. The procedure assumes the screen variables are initialised by the unload procedure. The program also assumes the load procedure will be used to save the new data.
PROCEDURE NAME: **LOAD.CMD**

FILENAME: remiss2.apx (binary)

PROGRAMMER: Neil Sumner

PURPOSE: Load data into REMISS database after editing.

VARIABLES: sn : Subproject number

scX#Y : screen X field number Y

DESCRIPTION: Procedure to save data from Rbase variables into REMISS database tables after editing is complete. The procedure replaces the data in the tables whether it was updated or not. The data is changed if it already existed in the table, otherwise it must be loaded.

---

PROCEDURE: **UNLOAD.CMD**

FILENAME: remiss2.apx (binary)

PROGRAMMER: Neil Sumner

PURPOSE: Read data from tables into screen variables for editing.

VARIABLES: sn : Subproject number

scX#Y : screen X field number Y

DESCRIPTION: Procedure to load data from tables into Rbase variables prior to editing. Also initialises optional variables which may be missing from a table.

---

**DATA EDITING PROCEDURES**

PROCEDURE NAME: **EDIT.CMD**

FILENAME: remiss2.apx (binary)

PROGRAMMER: Neil Sumner

PURPOSE: Allows user to edit data related to a particular Subproject.

VARIABLES: sn : Subproject number

scX#Y : screen X field number Y

DESCRIPTION: This procedure assumes the data is loaded into the appropriate variables. It also assumes any changes will be saved once editing is complete. The procedure allows the user to edit information using data entry screens.
PROCEDURE NAME: EDIT1.CMD

FILE NAME: remiss.apx (binary)

PROGRAMMER: Neil Sumner

PURPOSE: List all Subprojects.

VARIABLES: none

DESCRIPTION: Displays Subproject number and title for all Subprojects in database.

PROCEDURE NAME: EDIT2.CMD

FILE NAME: remiss.apx (binary)

PROGRAMMER: Neil Sumner

PURPOSE: Edit Subproject descriptive information.

VARIABLES: spn : Subproject number
          scX#Y : screen X field number Y

DESCRIPTION: This is a driver program for editing data from tables. Only tables containing descriptive information can be edited by this procedure. The procedure calls other procedures to read the data from the tables, edit the data and save the updated information in the database.

PROCEDURE NAME: EDIT3.CMD

FILE NAME: remiss.apx (binary)

PROGRAMMER: Neil Sumner

PURPOSE: Edit Appropriation Finances.

VARIABLES: spn : Subproject number

DESCRIPTION: Allows the user to edit the AEXPEND table for a particular Subproject. The Rbase EDIT command is used.

PROCEDURE NAME: EDIT4.CMD

FILE NAME: remiss.apx (binary)

PROGRAMMER: Neil Sumner

PURPOSE: Edit Contributory Finances.

VARIABLES: spn : Subproject number
DESCRIPTION: Allows the user to edit the CEXPEND table for a particular Subproject. The Rbase EDIT command is used.

PROCEDURE NAME: EDIT5.CMD

FILE NAME: remiss.apx (binary)

PROGRAMMER: Neil Sumner

PURPOSE: Edit Appropriation Staff Time Allocation.

VARIABLES: spn : Subproject number

DESCRIPTION: Allows the user to edit the ASTAFF table for a particular Subproject. The Rbase EDIT command is used.

PROCEDURE NAME: EDIT6.CMD

FILE NAME: remiss.apx (binary)

PROGRAMMER: Neil Sumner

PURPOSE: Edit Contributory Staff Time Allocation.

VARIABLES: spn : Subproject number

DESCRIPTION: Allows the user to edit the CSTAFF table for a particular Subproject. The Rbase EDIT command is used.

PRINTING FORMS

PROCEDURE NAME: ESTFORMS.CMD

FILENAME: remiss2.apx (binary)

PROGRAMMER: Neil Sumner

PURPOSE: Print forms for annual estimates.

VARIABLES: spn : Subproject number
             rpX#Y : page X field number Y

DESCRIPTION: The procedure first initialises the screen variables. Data is then loaded from the database into these variables. The forms are then printed. Extra pages are printed where a Subproject receives contributory funds. Totals are calculated where applicable. Rbase pointers are used to address report variables. These variables may be cleared between printing pages.
GENERAL PURPOSE PROGRAMS

PROCEDURE NAME: **ADHOC.CMD**

FILE NAME: remiss.apx

PROGRAMMER: Neil Sumner

PURPOSE: Menu option for REMISS command mode.

VARIABLES: none

DESCRIPTION: Exits the user from the REMISS menu program into Rbase command mode.

PROCEDURE NAME: **GETYEARS.CMD**

FILE NAME: remiss2.apx (binary)

PROGRAMMER: Neil Sumner

PURPOSE: Calculate present, past, and future financial years.

VARIABLES: pyear1-pyear4: previous financial years
cyear: current financial year
fyear1-fyear4: future financial years
prtblank: print blank forms (value from calling procedures)
%1: parameter sent (received from calling procedures)

DESCRIPTION: The procedure first calculates the current financial year from the date, depending on the part of the year and whether or not the current or next financial year is required. The four previous financial years and four future financial years are then calculated. These values are returned to the calling procedure as Rbase string variables of the format "yy/yy". This procedure uses Rbase pointers to refer to variable names.

PROCEDURE NAME: **MENU.CMD**

FILE NAME: Menu (binary)

PROGRAMMER: Neil Sumner

PURPOSE: To return user from command mode to the REMISS menu.

VARIABLES: None

DESCRIPTION: The complied should be available in the REMISS directory as a separate file.
PROCEDURE NAME: **NOTAVAIL.CMD**

FILE NAME: remiss2.apx (binary)

PROGRAMMER: Neil Sumner

PURPOSE: Dummy stub for menu options not implemented.

VARIABLES: None

DESCRIPTION: Displays a message to inform the user that the selected option is not available.

PROCEDURE NAME: **RBASE.DAT**

FILE NAME: rbase.dat (binary)

PROGRAMMER: Neil Sumner

PURPOSE: To invoke REMISS.

VARIABLES: none

DESCRIPTION: Checks clock on computer, then runs the REMISS application. Rbase looks for this file after loading. If this file is present it is immediately executed.

**MANAGEMENT REPORTS**

PROCEDURE NAME: **MREPORT1.CMD**

FILE NAME: remiss.apx (binary)

PROGRAMMER: Neil Sumner

PURPOSE: To print out Subprojects.

VARIABLES: none

DESCRIPTION: All the Projects along with their corresponding Subprojects and Subproject titles are printed by mreport1.cmd.

PROCEDURE NAME: **MREPORT2.CMD**

FILE NAME: remiss.apx (binary)

PROGRAMMER: Neil Sumner

PURPOSE: To print Subproject objectives.

VARIABLES: page : page counter
count : line counter
rsp1 - rsp8 : related Subproject numbers
ptr2 : pointer for the table PROJECT
ptr3 : pointer for the table RELSUBPJ

DESCRIPTION : Pointer #2 is set to the first Project number in the table PROJECT. The while loop is performed until all the Project numbers have been processed. A check of the line counter indicates whether or not a new page and headings are required. The setting of pointer #3 and the following 8 if statements is used to have all the related Subproject numbers of the current Subproject together, the command NEXT #3 ptr3 selects the next related Subproject number from the table RELSUBPJ, if no related Subproject number is found on any of the NEXT #3 ptr commands then any following if statements are not executed. Having determined the related Subproject numbers a report (MREPORT2) is printed.

PROCEDURE NAME : MREPORT3.CMD

FILE NAME : remiss.apx (binary)

PROGRAMMER : Neil Sumner

PURPOSE : To print out Subproject information.

VARIABLES : none

DESCRIPTION : Project and Subproject information printed out by the report MREPORT3.

PROCEDURE NAME : MREPORT4.CMD

FILE NAME : remiss.apx (binary)

PROGRAMMER : Neil Sumner

PURPOSE : To print estimate data entry forms.

VARIABLES : spn : Subproject number
ptr3 : pointer for the table PROJECT

DESCRIPTION : Accepts a Subproject number from the keyboard, if a Subproject number is entered then it is tested to see whether or not it exists in the database, if it does then the getyear.cmd program is run and the estimate forms are printed using the program mrept44&.cmd, if it doesn't exist then an error is displayed.

PROCEDURE NAME : MREPT44&.CMD

FILE NAME : remiss2.apx (binary)

PROGRAMMER : Neil Sumner

PURPOSE : To print estimate data entry forms.

VARIABLES : ptr2 : pointer for the tables
DESCRIPTION: By setting the pointer #2 in one of the tables mentioned above, information about the entered Subproject (in MREPORT4) is obtained and is placed into the appropriate variable. The NEXT #2 ptr2 command selects the next data from the current table, if there is no more data in the table for the current Subproject number then all the following if statements are not executed as the value of ptr2 will not be 0 and control is passed to the next SET POINTER command. Because of an RBASE constraint of allowing only 40 variables in a report the report had to be split into 4 parts.

PROCEDURE NAME: MREPORT5.CMD

FILE NAME: remiss.apx (binary)

PROGRAMMER: David Hawke

PURPOSE: To print Subproject staff and funding.

VARIABLES: page: page counter
count: line counter
pnold: old Project number
gtotal: Subproject total
gytotal: Project total
esp: existing Subproject
staffx: staff number x, used to hold staff names (values from profstaff.cmd)
classy: class number y, used to hold staff classifications (values from classif.cmd)
pz: print number z, used to show whether the class and staff have been printed (values from prtstaff)
staffct: staff count
pn: Project number
spn: Subproject number
prct: print count (value from prtstaff)
pitle: Project title
curtable: current table

DESCRIPTION: Pointer #2 is set to the first Project number in the table PROJECT. The while loop is performed until all the Project numbers have been processed. The variable esp is set to an existing Subproject number so as Project totals can be printed when a change in Project number is detected. A check of the line counter will indicate as to whether or not a new page is required. A comparison of the current Project number (pn) and the old Project number (pnold) will determine whether or not a new Project number heading is required, by setting the pointer #2 in the table PROJITTL we can obtain the current Project title (pitle), the check on the variable count (IF count > 0) is so that if report headings have just been printed then there is no need to skip 2 lines. The curtable variable is set so as to use the same to calculate the current year funding (curnyr.cmd). The previous year's funding is calculated by using the dates supplied by getyears.cmd. The variable cyear is reset to the current year otherwise the current year professional staff would be incorrect (profstaff.cmd). By setting the printnum to 1 and running the program prtstaff.cmd we can obtain the first professional staff member, and have that persons name printed on the first detail line by the report MREPORT5. By changing the printnum value we can print the remaining professional staff
(prtstaff). The only way to have an accurate line count record was to increment the line count (count) by the value of prtct and adding 1 for the detail line.

PROCEDURE NAME: CURNTYR.CMD

FILENAME: remiss2.apx (binary)

PROGRAMMER: David Hawke

PURPOSE: Calculate current and previous years funding.

VARIABLES: pfun ds: provision funds
           sfunds: supplement funds
           funds: in some cases total of pfun ds and sfunds
           cyear: current year (value from mreport5.cmd)
           curtable: current table (value from mreport5.cmd)
           ptr3: pointer for the table held in curtable

DESCRIPTION: By setting the pointer #3 we can determine whether the funds value held in the table are provisional, supplement or forecast, and then calculate the appropriate figure, and adding this amount to the year total (ytotal).

PROCEDURE NAME: PROFSTAFF.CMD

FILENAME: remiss2.apx (binary)

PROGRAMMER: David Hawke

PURPOSE: To find the professional staff.

VARIABLES: curtable: current table (value from mreport5.cmd)
            curssname: current surname
            staffct: staff count
            cyear: current year (value from mreport5.cmd)
            classif: classification
            ptr1: pointer for the table held in curtable
            ptr3: pointer for the table salaries
            stffx: staff number x, used to hold staff names
            classy: class number y, used to hold staff classifications (values from classif.cmd)
            rank: rank (value from classif.cmd)

DESCRIPTION: The pointer #1 is positioned at the first record meeting the specified condition, and the loop is continued until all the records meeting the condition are processed. The classification of cursname is taken from the table salaries held in pointer #3. The classif.cmd program is run giving professional staff involved in the current Subproject a particular number. Depending on the staff count each professional staff member is placed into a stffx and classy variable.
PROCEDURE NAME: CLASSIF.CMD

FILENAME: remiss2.apx

PROGRAMMER: David Hawke

PURPOSE: To classify professional staff.

VARIABLES: rank: staff ranking value (1-9)
        classif: classification (value from profstaf.cmd)

DESCRIPTION: Each professional staff member of the current table is giving a rank value depending on their classification.

PROCEDURE NAME: PRTSTAFF.CMD

FILENAME: remiss2.apx

PROGRAMMER: David Hawke

PURPOSE: To print out the professional staff in rank order.

VARIABLES: curtable: current table (value from mreport5.cmd)
        ptr1: pointer to the table held in curtable
        rankct: rank count
        prct: print count
        printnum: print number
        staffct: staff count
        staffx: staff number x, used to hold staff names
        (values from mreport5.cmd)
        classy: class number y, used to hold staff classifications (values from mreport5.cmd)
        pz: print number z, used to show whether the class and staff numbers have been printed
        pstaff1: print staff 1, used by the report MREPORT5 to print the first professional staff member on the detail line.

DESCRIPTION: Compares the classy to the rankct if they match then depending on the value of printnum (printnum = 1) then the staffx value is placed into the pstaff1 value and printed out with the detail line, otherwise providing printnum = 2 and pz = 0 the staffx will be printed on a separate line.

PROCEDURE NAME: MREPORT6.CMD

FILENAME: remiss.apx (binary)

PROGRAMMER: David Hawke

PURPOSE: To print all the staff's current Subproject commitments.

VARIABLES: page: page counter
        count: line counter (value from prtproj)
        ptr2: pointer for the table SALARIES
currsname : current surname
curtble : current table
curfinds : current funds
prt : print, (value from prtproj.cmd)

DESCRIPTION : Getyears.cmd is run to determine the current year. Pointer #2 is set to the first name in the table SALARIES. A check on the line counter (count) is made to see whether a new page and headings are required. By calling the program prtproj.cmd we can look at the current table (curtble) and print all the Subprojects of the current staff member. If there were no Subprojects found then the 2 separating lines are not printed.

PROCEDURE NAME : PRTPROJ.CMD

FILE NAME : remiss2.apx (binary)

PROGRAMMER : David Hawke

PURPOSE : To print all the staff current Subproject commitments.

VARIABLES : ptr3 : pointer for the table curtble
curtble : current table (values from mreport6.cmd)
spn : Subproject number
pn : Project number
prtime : print actual time
currsname : current surname (value from mreport6.cmd)
count : line counter

DESCRIPTION : The pointer #3 is set in the current table and all the Subprojects for the current staff member are processed, the Project number is taken from the table PROJECT using the current Subproject number (spn). The detail line is then printed by MREPORT6.

FINANCIAL REPORTS

PROCEDURE NAME : FREPORT1.CMD

FILE NAME : remiss.apx (binary)

PROGRAMMER : Neil Sumner

PURPOSE : To print operating expenditure estimates.

VARIABLES : spn : Subproject number
ptr3 : pointer for the table PROJECT

DESCRIPTION : Accepts a Subproject number from the keyboard, if the Subproject number exists then the current year is returned by a called program (getyears.cmd). The operating expenditure estimates are then sent to the printer by the called program freprt1.&.cmd. If the Subproject number does not exist then an error message is displayed.
PROCEDURE NAME: FREPORT2.CMD

FILE NAME: remiss.apx (binary)

PROGRAMMER: Neil Sumner

PURPOSE: To print operating expenditure estimates.

VARIABLES: pn : Project number
           spn : Subproject number
           ptr3 : pointer for the table PROJECT

DESCRIPTION: Pointer #1 is set to the first Subproject number in the table PROJECT, and the current Project and Subproject numbers are displayed on the screen, Project operating expenditure estimates are printed by the called program frept1&.cmd, the current year is determined by the calling of getyears.cmd. This process is continued until operating expenditure estimates have been printed for all Subprojects.

PROCEDURE NAME: FREPRT1&.CMD

FILE NAME: remiss2.apx (binary)

PROGRAMMER: Neil Sumner

PURPOSE: To print operating expenditure estimates.

VARIABLES: code1 : account code value (pointer)
           code2 : justification of expenditure (pointer)
           cyear : current financial year (from getyears.cmd)
           spn : Subproject number (from FREPORT1/FREPORT2)
           total : total of all account code values
           ptr2 : pointer for the tables AEXPEND/CEXPEND

DESCRIPTION: Pointer #2 is set to the current Subproject number of the current year in the table AEXPEND. By creating variables the current value of code1 (a variable itself) takes on the value of estimate in pointer #2, and the variable held in code2 holds the value of justify in pointer #2. A running total is kept in the variable total. This process is performed until the loop condition is not met, then all the details of the selected/current Subproject are printed. Because of an RBASE constraint of allowing only 40 variables in a report the printing had to be split into two reports. All variables are then cleared and the current year (cyear) is determined by the program (getyears). A test to see if any contributory funds exist in the table CEXPEND is made, and if there are then, the same steps are taken as in the table AEXPEND, otherwise control is returned to the calling program.

PROCEDURE NAME: FREPORT3.CMD

FILE NAME: remiss.apx (binary)

PROGRAMMER: David Hawke

PURPOSE: To print Subproject funding.

VARIABLES: page : page counter
           count : line counter
pold : previous Project number
atotal : total of appropriation funding
ctotal : total of contributory funding
gatotal : grand total of appropriation funding
gctotal : grand total of contributory funding
pn : current Project number
spn : current Subproject number
afunds : appropriation funds
cfunds : contributory funds
ptitle : Project title
fsour : funding source
ptr2 : pointer for the table PROJECT

DESCRIPTION : Pointer #2 is set to the first Subproject number of the first Project. The variables pn and spn are set to the values held by project# and subproj# in pointer #2. A check for a control break on the Project number field (pn) and if there is a change in this field then totals are printed using freprt3&.cmd and the variables atotal and ctotal are reset. If the count variable is greater than 29 then new page and column headings are printed, and the count and page variables are altered. If a change in Project number is detected then a new Project title heading is printed. Afunds and and cfunds are calculated from their respective tables (AEXPEND/CEXPEND) and are then added to the their total fields (atotal/ctotal). A check for contributory funds on the current subproject number is made and depending on the result held in pointer #3 depends on the value place into fsour. The report details are then printed out for the current Subproject number by freprt3.cmd. And the next Subproject number is taken, this process continues until the loop condition is not met. The value of the grand total variables (gatotal/gctotal) is placed into the variables atotal and ctotal so that the same report (FREPTRT3&) could be used for printing them out.
APPENDIX B
REMISS MENU STRUCTURE

REMISS - MAIN MENU

(1) System Description
(2) Data Entry
(3) Edit Subproject Information
(4) Print Management Reports
(5) Print Financial Reports
(6) Print Data Entry Forms
(7) Database Administration
(8) Perform Ad Hoc Query
(9) Exit

REMISS SYSTEM DESCRIPTION

(1) Introduction
(2) Data Entry
(3) Editing Subproject Information
(4) Management Reports Available
(5) Financial Reports Available
(6) Introduction to Command Mode
(7) Exit

DATA ENTRY

(1) Enter New Subproject Forecast
(2) Enter Annual Estimates
(3) Exit

EDIT SUBPROJECT INFORMATION

(1) List Subprojects
(2) Edit Subproject Descriptive Information
(3) Edit Appropriation Finances
(4) Edit Contributory Finances
(5) Edit Appropriation Staff Allocation
(6) Edit Contributory Staff Time Allocation
(7) Exit
PRINT MANAGEMENT REPORTS

(1) Print Inventory (All Subprojects)
(2) Print Objectives and Interrelations (All Subprojects)
(3) Print Statistics (All Subprojects)
(4) Print Descriptive Information (one Subproject)
(5) Exit

PRINT FINANCIAL REPORTS

(1) Print Operating Expenditure Estimates (One Subproject)
(2) Print Operating Expenditure Estimates (All Subprojects)
(3) Print Subproject Funding (All Subprojects)
(4) Exit

PRINT DATA ENTRY FORMS

(1) Print Initial Forecast Forms
(2) Print Annual Estimate Forms (One Subproject)
(3) Print Annual Estimate Forms (All Subprojects)
(4) Exit

DATABASE ADMINISTRATION

(1) Update Subproject Status
(2) Update Subproject Application Area
(3) Update Abbreviated Title
(4) Update Project Groups
(5) Update Staff Classification and Salaries
(6) Exit