NAME

s.db.rim - RIM data base management/query interface for GRASS sites data.
(GRASS Sites Management Program)

SYNOPSIS

s.db.rim
s.db.rim *data_base*

DESCRIPTION

For some informations on LINUX and RIM look at the BUGS section

s.db.rim allows users to create, manage and query information about site locations (sites) across the landscape. Required inputs can be entered interactively, or from the command line. Command line input may be entered through a prepared text file or from the keyboard (standard input). The *s.db.rim* command language is defined in SECTION ONE below. The menu-driven interactive version is described in SECTION TWO of these manual pages.

These programs are actually a marriage of the GRASS environment and the programmer's interface library of the relational data base management program RIM, distributed publically by the University of Washington Academic Computing Services as FORTRAN 77 code. Your system must have a FORTRAN 77 compiler to use *s.db.rim*.

SECTION ONE -- THE COMMAND VERSION

The command-line driven version of *s.db.rim* is run by typing the below command, where *data_base* is the name of an existing RIM sites data base:

s.db.rim *data_base*

The sites data bases are stored in a subdirectory named 'rim/sites' in the user's current mapset. Data bases in other mapsets, selectable through the GRASS *g.mapsets* command, can be accessed for 'read-only' retrieval of records. Each mapset may have many data bases. Each data base within a mapset must have a different name; user-supplied names for data bases are limited to seven (7) characters in order to maintain compatibility with the standard version of RIM. As with other GRASS commands, mapsets are searched in the mapset SEARCH_PATH order when a data base needs to be opened.

Each site data base is composed of multi-field records (rows or tuples, in DBMS jargon). Each field and its position in the site form is defined via input to the .make command when a data base is originally defined. It is possible to add new fields or change the length of existing fields after data has been loaded, however this is not straightforward and is not described here; deleting of fields is also possible, but requires even more experience and knowledge. The user needs to carefully design the data base fields and form (layout) and check the operation with a few pieces of test data before loading data for a large number of sites.

COMMANDS

(Note: For each of the "dot commands", i.e., .make, described below there is a menu choice to selected when running the interactive version. The interactive menus are described in the SECTION TWO of this document. Some display capabilities exist in the interactive version which are not directly implemented in the command version.)

The commands are given alphabetically here for easy reference. The .make command is required to create a data base and, therefore, will be the first to be entered by a new user. Abbreviations down to the string shown in ( ) are accepted; this is primarily for those giving *s.db.rim* commands from a terminal, but abbreviations may also be used in batch files.

Each command is introduced with an input record (line) which starts with a period and is followed by one of the words shown below; for some commands the command line also contains one or more required or optional parameters. Additional or optional input instructions/data for a command are supplied on successive lines; a .end line is needed by some commands to signify the end of these input lines.

Alphabetical Command Summary

"*!command*"

This is the only *s.db.rim* command not starting with a period. "command" is a single shell command line which is executed by a "G_system()" call (see GRASS gis library). Many UNIX utilities (e.g., *vi, ls, print*) and most GRASS commands (e.g., *d.rast, d.points, g.list, g.region*, *d.zoom, r.mask*) may be executed. It is permitted, and often useful, to change "region" and "MASK" within *s.db.rim.* Multiple commands may be separated by ";" in the standard UNIX way. Note that a "!cd directory; ls" will change to the specified directory and list files, but the effective working directory for *s.db.rim* will not be changed when the command terminates.

"*.add (.a)*"

Add a new site record (row) to the open data base. Each line following contains a field name followed by spaces and/or tabs then the value or character string to store for that field. Field information lines end with .end. Some fields may be absent and fields may appear in any order. Checks are made for the input of data for the one required field (site number), for string length for string type fields, and for duplicate site numbers. If split fields are used in the data base layout (see .make), text data for each split field must be added as a separate line. If there are any problems, the record will not be stored and a message will be output. This format makes it relatively easy to import data from most other DBMS. The ".print -a" command, see below, outputs data in this list format.

```
Example:

 .add
 site_id   204
 north     4690673.30
 east      601410.00
 reference Jones (1987)
 .end
```

"*.backup (.b) file_name*"

The .backup command is used to dump the entire data base from the RIM binary files to a text file format (see UNLOAD in the RIM User's Manual). The file_name can be a relative path name or full path name. The file will contain the data base definition, screen layout information, and

tabular data. This text file is transportable to RIM or *s.db.rim* running on any other computer; it may also be reloaded to recreate the *s.db.rim* data base. A message will be output if there is any problem writing the .backup file. Backup can only be done on data bases in the user's current mapset.

To reload your data base from the backup file (normally not necessary):

```
GRASS 4.1> cd $LOCATION/rim/sites #right directory
GRASS 4.1> rm db_name.rimdb1      #remove data base (or mv to somewhere)
GRASS 4.1> rm db_name.rimdb2      #remove data base (or mv to somewhere)
GRASS 4.1> rm db_name.rimdb3      #remove data base (or mv to somewhere)
GRASS 4.1> rim                    #run RIM manually
RIM> input "path/file"            #RIM rebuilds data base from data
                                   written by .backup
RIM> exit
```

"*.change (.c) [-l]*"

Without the "-l" flag, each line following .change is in the same format as for the .add command. The site number field is required and the site number must match an existing site in the data base. Only those fields for which lines are provided are changed in the record. After the .end the changed record is stored, if all is ok, otherwise a message is output.

If the "-l" flag (for "list") is given, the site number field is omitted and the specified field values are changed for all sites currently selected by .find and/or .query.

"*.delete (.d)*"

This command is used to delete data records for sites. Deletion of sites is permanent. A backup of the data base, or copies of the data base files, are the ways to protect your valuable data.

The lines following the .delete command should contain only the site numbers, with a .end line being last.

The following command sequence will delete all the sites currently on the internal site list (the result of the last .query or .find command) after asking for approval.

```
.delete
.end
```

"*.end (.e)*"

Ends multi-line input for several other commands.

"*.exit (.ex)*"

Use .exit to end operation of *s.db.rim* cleanly. In general, do not use CTRL-C to exit unless absolutely necessary. When .exit is encountered in a batch file, input will revert back to the previous file, or the terminal, if any, which called the batch file.

"*.find (.f) [-a | -d] [-m] [-r]*"

The .find command is used to find the site(s) closest to a given point (the target). The target can be defined in one of several ways. The found sites are stored on an internal sites list for output by other commands; however, see note 2, below. The found sites are stored on the internal sites list in order of proximity to the target location.

The optional .find command line parameter specifies the current MASK (-m), if any, or the current region (-r), as a filter on the retrieved sites. -m automatically implies -r, as the MASK is not defined

outside the current region. If the -a flag is given, the retrieved sites will be appended to those previously retrieved with a .query or .find; duplicates will be automatically discarded. The -d flag causes the retrieved sites to be deleted from the internal site list, if present there. Very complex selections can be done by interspersing appends and deletes to arrive at a final list of sites. For instance, selecting those sites within 2000 meters of a target and then deleting those within 1500 meteres of the target will give a final list of those from 1500 to 2000 meters.

The single required line following the .find line gives the program the necessary target information. The following examples show the possibilities.

```
find> 602793.90 4379010.00
```

will find the one site nearest these coordinates and store it on the internal site list.

```
find> 619840 4599000 10
```

will find the 10 sites (or fewer, if there are not that many) closest to the given location.

```
find> site 132 10
```

will find the 10 sites closest to the location of site 132 in the data base (including site 132). If site 132 does not exist, no action is taken.

```
find> distance from 472910.06 5732001.0 5000
```

will find all sites within 5000 (meters, in UTM or Lat-Long coordinates) of the target location.

```
find> distance from site 16 -2500
```

will find all sites greater than 2500 (meters) from the location of site 16.

Notes for .find:

1. All sites found are stored on the site list in order of proximity to the target location (sorted by distance from target).

2. The number of sites found is automatically printed to the active output device/file.

3. If mask is specified, the effective region is automatically set to the current region (because the GRASS mask is only defined for the current region).

4. Region and mask filtering uses the current resolution for the region to test if a point falls within a cell in the masking map.

5. In the last two examples the string "distance from" must be exactly matched. Also, the word "site" must be exactly matched.

6. If the "distance from" radius is given as a negative value, points outside the target circle are selected; whereas, if a positive value is given, points inside the circle are selected.

7. The current region may be changed with !g.region or !d.zoom prior to doing a .find, and the mask may be set or removed with a variety of GRASS commands.

8. The "find>" prompt is given only when input is from a terminal.

"*.help (.h)*"
Prints a help screen to the output device or file. Useful to have when using *s.db.rim* from a terminal, or when writing a script file of commands.

"*.input (.i) [file]*"
The lines in the file given are read and processed as commands or data until an end of file is reached or until a .exit command is found. Input files may call other input files to a nesting depth of eight. Without a file name, stdin is used as the input file.

"*.list (.l)*"
Lists the available data bases in the current mapset search path.

"*.make*"
Using the .make command you create a new data base in the current mapset by specifying the following items which define the screen (page) layout for displaying and printing the site records, as well as the information fields:

1) The fixed text part of the screen layout.
2) The positions, types, names and lengths of data fields.

Three fields must always exist in a data base; each of these field types may only occur once in a data base layout:

1) Type ’s’ Site identification number field (an integer).
2) Type ’x’ Easting coordinate of the site (a double float).
3) Type ’y’ Northing coordinate of the site (a double float).

The other field types, which may occur in any combination and order, are:

4) type ’i’ An integer field.
5) type ’f’ A double precision float field.
6) type ’t’ A text field.

Each of the fields can be positioned anywhere within the screen layout, which has a limit of 19 lines by 80 columns. A maximum of 70 fields may be defined within this space. A field is specified in the screen layout by a tilde (~), a field type character, a field name and enough trailing tildes to fill out the desired field length.

Each line following the .make command is taken to define a line of the screen layout until a .end is reached. If a mistake is made on any of the input lines, the .make will fail. The .make information may be prepared in advance as a text file (this facilitates fixing mistakes) and the .input command can be used to read in this file. An example text file for a data base screen layout follows, with some important explanatory notes.

```
.make
              Archaeological Sites Database
              ============================
```

```
Site #: ~sSite~~~      Entered By: ~tEnter_by~~~~~~~~
Description:                    C-14 Date:  ~iAge~~~
    ~tDescript.1~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    ~tDescript.2~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    ~tDescript.3~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Type:  ~tType~~~~~~~~~~~~~ (Should be Arch. or Hist.)
Date:  ~tEnter_Date~~~~~~~      Square Miles: ~fArea.4~~~
North: ~yNorth~~~~~  East: ~xEast.1~~~~
.end
```

Notes:

1) Any text not preceded by a tilde (~) character is taken to be part of the constant or fixed text portion of the form.

2) A field definition begins with a tilde (~) character immediately followed by a single character which indicates the data type of the field (s,x,y,i,f or t). Immediately following the data type character is the field name of 1 to 16 characters. Field names can be composed of any characters from the following set: [A-Z,a-z,_,0-9]; the RIM program and library do not distinguish upper and lower case in field names, so you should avoid making names which differ only in case. Field names may not begin with a numeral [0-9]. The rest of the field length is padded with tilde (~) characters to the required maximum length.

3) The minimum field width is three characters; e.g., "~tA". Be sure field widths for all fields are wide enough for the values and strings you expect to store there; e.g., UTM northings require at least 11 spaces.

4) For text fields it is possible to continue a field across more than one line. This is done by appending a .1 to the field name forming first portion of this "split field", a .2 for the second portion, etc. This text field splitting affects how information is organized for input and output; the composite text string is concatenated (unused portions of fields are retained as spaces) and treated as a unit for storage and queries to the data base.

5) For the double precision floating point fields (types x, y and f), the number of decimal places to print may be specified by appending ".n" to the field name, where n is the number of decimals places required. Values of n from 1 to 12 may be used. Two decimal places is the default if ".n" is not specified. Be certain that you make the field wide enough to print the integer and decimal portions of the values that will be stored in the field; include space for a sign and the decimal point. (If it is desired to print zero decimals, whole numbers, use an "i" type field.) In the example above, the northing coordinate (y) would be output with two decimals, the easting (x) would have only one decimal place, and the "Area" would be printed with four decimal places.

"*.output (.o) [file or | process]*"

Causes all output (except some error messages) from *s.db.rim*, including that from the .print command, to go to the named path/file (may be a full or relative path name), or to be used as standard input by the process (a pipe). If no parameter is given, output returns to stdout, usually the user's terminal. An example of the pipe usage would be

```
.output | grep "easting" | wc -l > /tmp/my_count
```

A pipe is closed whenever the .output command is given again, or on a .exit command.

"*.pack (.pa)*"

This should be used when numerous data records have been deleted to recover disk space in the RIM binary data base files. It works by doing a .backup to a temporary file; moving the data base files to new names (*.bakdb*); running RIM to rebuild the data base; and, if the rebuilt data base can be opened and read, the temporary files are deleted. The user is informed if this process fails. Packing can only be done on an open data base located in the user's current mapset.

"*.print (.p) [-a | -l]* "

This command outputs the full site records for the sites currently stored on the internal sites list (result of last .query or .find). Without the flag, the screen layout format is used. With the -l flag, for list format, the field name followed by the contents are output one field per line. The -a flag also outputs in the list format but with a .add line and a .end line surrounding each site record printed; data files in this form can be read with .input, thus they form one kind of backup mechanism and can be used to transfer data (not the data base layout) from one GRASS system to another. The destination for the output is set by a previous .output command (default is stdout).

"*.query (.q) [-a | -d] [-m] [-r]*"

The .query command is used to retrieve sites via an SQL-like request to RIM, including a user specified "where clause." All fields for each site meeting the selection criteria are retrieved.

The optional .query command line parameters cause points not in the region (-r) and/or mask (-m) to be rejected, so these conditions need not be tested in the "where clause." The -a flag causes the retrieved sites to be appended to those previously retrieved by .query or .find; duplicate entries are automatically discarded. The -d flag causes selected sites to be deleted from the current list, if present.

After the query command line, any number of lines may be entered to define the SQL "where" clause. A .end line is required to finish the request and begin data retrieval. See examples below.

The "distance from" clause may also be used as additional selection criteria exactly as described in the examples and notes for .find. It must be entered as a separate line to the query prompt.

The retrieved records may be printed at time of retrieval, rather than after the completion of the query command by including a .print (.p) line with the same options for print format as in the .print command (see above); e.g. .p -a to output in the "list add" format. The .print clause must be entered as a separate line to the query prompt. This feature is most useful when working with very large data bases where retreval time is significant. See example 2 below.

```
Example 1

query> where density < 20 and (date = "10/14/89"
query> or county eq "San Marcos")
query> .end


Example 2

query> where east <600000 and name like "*Jones*"
query> distance from site 12 3000
query> .print –a
query> .end


Example 3

query>.end
```

The where and distance from clauses are each optional. If both are omitted, only the mask and region on the .query command line restrict the search; if mask and region are also omitted, all sites will be retrieved (Example 3). When querying for sites the where clause is processed first, the current region and mask tested next (if requested), then the distance from clause is applied; a site must pass all tests to be put on the internal site list for output by other commands.

Notes: (Also see Notes for .find)

1. The retrieved sites are stored on the internal site list in the order returned from the data base by RIM, not necessarily in site number order or the order the data was loaded. A "distance from" clause results in a final sorting by proximity to target.

2. See the RIM User's Manual for additional information on the "where" clause in the "select" command, especially the quotes required for matching character string fields, and the allowed comparison operators.

3. In the where clauses of the examples, "density", "date", "county", east", and "name" are field names (column names in RIM) defined when the user initially makes the data base.

4. Each .query or .find resets the internal site list (even unsuccessful ones), unless the -a or -d flags are used.

"*.read_site (.re) site_list [comment_field]*"

This command reads an existing GRASS site list and creates a data base record for each site. If the comment or description field of all entries in the site list begin with # and a number, the number becomes the site number in the data base. If some of the sites in the GRASS site list do not have a # at the beginning of the comment field, the sites are numbered sequentially starting with 1. (These options are similar to the way the GRASS sites-to-raster [in s.menu] works.)

if a data base field name "comment_field" is entered on the command line, the comment will be stored in that field for each site. If an integer or float field is specified, and attempt is made to interpret the comment as that type of number; if this interpretation fails, 0 or 0.0 is stored.

If the site number duplicates one already in the data base or found earlier in the site list, it is not added.

Once the sites have been loaded by .read_site, use .change (or the interactive version) to add data to other fields for those sites.

"*.remove*"

This command, which requires a "y" as confirmation on the next line, entirely removes the three binary files which constitute your RIM data base. Use with care. Backup files must be removed individually by the user, if desired, from the $LOCATION/rim/sites directory.

"*.show (.sh)*"

This command is used to output the screen or page layout as defined for the current data base. It serves as documentation of the data base definition and as a reminder for field names, types and lengths. By using an editor to surround the output of .show with .make and .end lines, it can be used to reload the data base definition with .input.

"*.site_list (.si) file_name [field_name]*"

This command writes the site locations and the site numbers to the specified file in the site_list directory in the current mapset. If the file exists, the sites are appended to the current list, otherwise, a new site list file is created. A "field_name" may be optionally specified; if so, the contents of that field (retrieved from the appropriate site record) are inserted as the comment (following a '#') in the site list. The site number is used if no field name is supplied.

A comment line is inserted in the site_list file with the current date and time and the name of the data base producing the site locations. The format used for each site is:

```
easting|northing|#comment
```

"*.tables (.t)*"

Prints the table structure of the currently opened RIM data base. This is the same output generated by a "list *" command when running RIM manually. The information for the table named "data" is useful for review of the user's field definitions. The information for the two other tables is for internal use by *s.db.rim.*

SECTION TWO -- THE INTERACTIVE VERSION MENUS AND COMMENTS

SYNOPSIS

s.db.rim

DESCRIPTION

The interactive version of *s.db.rim* allows you to create, manage and query information about site locations (sites) across the landscape. Operations are done on a data base through a series of menus explained below. Most of the menus use VASK screens; the user should become familiar with keys that move the cursor among the fields to be entered (RETURN, ENTER, CTRL-L, CTRL-K, etc.).

THE MAIN MENU

Below is the main menu. Option 1 is the default. Note the status line at the top of the menu, and the fact that 8 records have been selected by the latest find or query operation (between items 2 and 3). Note, also, that CTRL-C can be used to exit from this menu (and most other menus in the program) back to the GRASS prompt. The specifics of each menu choice are described below.

```
   s.db.rim              MAIN  MENU                  Version 1.4
       Data base <water> in mapset <rono> open.  25 records.

      1  Open a data base
      2  List available data bases
--------  Retrieve/Output Site Records (8 currently)  --
      3  Find sites in proximity to a Target point
      4  Query to select site records (SQL)
      5  Show selected site records on Terminal
      6  Display maps/selected sites on graphics terminal
```

```
     7  Output selected site records to Printer or File
     8  Create a site_list from selected records
------------  Add/Edit Site Records  ----------
     9  View a single site record
    10  Add a site record
    11  Change a site record
    12  Delete a single record or all selected records
------- Other functions -- Shell Command -- Exit ---------
    13  Make a new data base & Management Functions
    14  Execute a shell command

     0  Done -- Exit from s.db.rim

  AFTER COMPLETING ALL ANSWERS, HIT <ESC> TO CONTINUE
         (OR <Ctrl-C> TO EXIT THIS PROGRAM)
```

1. Open a data base. If a data base is already open, it is closed before the requested one is opened. Only data bases in the user's current mapset may be modified; others are opened in read-only mode; this will be indicated on line 2.

2. List available data bases. For each mapset in the current GRASS mapset search path, the names of the existing data bases are listed.

3. "Find" sites in the data base relative to a specified target location. This is used to select sites based on proximity to the target and, optionally, sites within the current region and, optionally, sites falling in active cells within the current GRASS mask. Two modes of targeting are provided: the N sites closest to the target, and all sites within (or outside) a circle of specified radius from the target. The FIND/QUERY TARGET MENU discussed below accepts region/mask/target specifications from the user. The selected sites are then displayed one at a time until CTRL-C is entered; then other operations, choices 5-8, can be done with these sites. The line on the menu between 2 and 3 shows the number of sites currently selected by choices 3 or 4.

4. "Query" sites in the data base using an SQL-like "where clause," including specifications for region/mask/target (circle only) as in 3, above; see FIND/QUERY TARGET MENU section below. The where clause can test for ranges or matches for numeric data base fields, or matches on full strings or substrings for text fields. The selected sites are then displayed one at a time until CTRL-C is entered; then other operations, choices 5-8, can be done with these sites. This clause is entered on a menu described below; see QUERY COMMAND MENU section, below.

The where clause may use parentheses ( ) to control the order of comparisons. Field names are not case sensitive within where clauses. The following comparison operators are valid for all types of fields:

```
eq   or   =          ne   or   <>
ge   or  >=          le   or   <=
gt   or   >          lt   or    <
```

String comparisons are case sensitive and are done character by character. Substrings comparisons may be done with the "like" operator as in:

```
where name like "*Jones*"
```

Note that the string being tested against the name field for each record is in quotes (single or

double) and that wild card comparisons can be done in the standard way with '*' and '?' characters.

Logical comparisons may also be combined with those operators above. The permitted logical operators are:

```
and       or       not
```

The following complex example should be examined. The line breaks can occur between any tokens (words, values, operators), except within quoted strings.

```
where (name like "*Jones*" or name = "Smith")
and ( ( site < 300 and not (site = 251 or site eq 15) )
or east < 601000 )
```

5. This choice will display the site records resulting from the last find/query one at a time on the terminal. Use ESC or enter a number to display another record and CTRL-C to end the display.

6. If a graphics monitor is active, the locations of the selected sites will be displayed. The user may choose to erase the screen; display raster, vector, and/or site maps; or display the selected sites from the data base. These maps are requested through the following interactive screen. Just enter ESC to skip this step. If no data base sites are currently selected, that section of the menu will not appear; but the menu can still be used to display the other types of maps. This display function is a major added function of the interactive version of the program; display is not so easy in the command version.

```
                SELECTION MENU FOR ITEMS TO DISPLAY


Enter raster and/or vector map names, if desired


 _______________  Raster map to display
 _______________  Vector map to display in color: __________
 _______________  Site list to display
                  Dpoints with: size=3_ type=box____ color=white____
                _ Display currently selected sites (enter x)
                  d.sites with: size=6_ type=x______ color=red______
                _ Erase graphics screen (enter x)
                  d.erase  black____


              AFTER COMPLETING ALL ANSWERS, HIT <ESC> TO CONTINUE
                           (OR <Ctrl-C> TO CANCEL)
```

7. This selection results in a screen prompting for the name of the file to output the selected site records to, and for optional formatting selection. If the file name is lp, the site records are sent to the printer. The optional formatting choices are for export of data in list format (see .print in the first part of this manual page for *s.db.rim* for information and examples).

8. Using this choice you can write (or append) the currently selected sites to a GRASS site_list file in your current mapset. A short menu prompts for the name of the site_list file, and also for the name of a field to be used for the "comment" in the site_list (the site number is the default field). The current date and time, and the names of the mapset and data base in use are entered as an

information line in the site_list file. Note that various kinds of raster map layers can be produced from a *s.db.rim* data base by writing site_lists with different fields as "comments" then converting the site_lists to raster files with *s.menu*.

9. Choices 9-12 operate on only a single site and do not use or modify the internal list of sites selected by find/query (choices 3 or 4). Choice 9 is the way to view a single site record, selected by site number. After viewing, ESC will allow entry of another site number and CTRL-C will exit to the main menu.

10. Use this selection to add a new site record to the data base. (A new site is one whose site number does not currently exist in the open data base.) After making this selection, the data base layout will be displayed and you should enter the available information appropriate to each field; the only required entry is the site number field. If values for numeric fields are not entered, zero values will be stored. Unused portions of text fields are stored as strings of spaces.

11. After making this selection and specifying the site number to change field information for, the data is entered as for choice 10, except that the site number cannot be changed. (The command version of the program has provision for making bulk changes after a find or query; see .change.)

12. To delete a single record, enter its site number when requested. All site records chosen by the last find/query operation may be deleted by entering "list" in place of the site number. BE CAREFUL with this, *deleted records are really gone.*

13. This choice starts a new menu with less commonly used functions. See MANAGEMENT MENU section below.

14. The program will prompt you for one-line Shell Commands until you enter just a <RETURN> to return to the main menu. Often useful for changing the GRASS region, setting a MASK, etc.

FIND/QUERY TARGET MENU

This is the screen to set up the region/mask/target information for the find choice (3) and the query choice (4), except that item B is omitted for choice (4). The choice to append or delete selected records will only be given after a successful find or query has stored some records on the internal record list. See .find and .query for more information.

If a graphics monitor is not active, the "mouse" item is omitted from the screen; and, if a mask is not set, that choice is omitted. The choices entered on this example screen will result in all the sites within a 1500 (meters) radius of the target point (to be chosen with the mouse) being selected and stored on the internal site list by find or query. They are stored in order of proximity to the target. If a site is used as the target, it is always the first in the retrieved list (useful for just selecting one site by number). If a mouse is chosen to select the target point, a menu to display reference maps is presented, exactly as in choice (6), prior to actually activating the mouse.

```
        QUERY/FIND:  REGION/MASK/TARGET SELECTION MENU
  Data base <arch> (READONLY) in mapset <PERMANENT> open.  113 records.
    Mark requests with 'x' and enter required values.

               Respect current region   _
               Respect current MASK     x
              (forces current region)

A.  Find all sites within (or outside) a circular target   x
```

```
             and give the radius (negative for outside)  1500.00_____
                         OR
B.  Find a number of sites nearest a point   _
         and the number of sites requested   _________

    After selecting A or B, complete one(!) of these:
        1. x to select target point with mouse    x
        2. Enter site number for target point     __________
        3. Target coordinates              east   0.00_________
                                          north   0.00_________

 Append/Delete to current FIND/QUERY site list (a | d) _
 Reset to default choices for this menu _

         AFTER COMPLETING ALL ANSWERS, HIT <ESC> TO CONTINUE
                       (OR <Ctrl-C> TO CANCEL)
```

QUERY COMMAND MENU

The following screen completes the information for a query (choice 4). It may be left blank if no "where clause" is required. After a successful query, the selected records are displayed one at a time by hiting escape; CTRL-C will quit the display and return to the main menu where several choices of operation on the retrieved sites are offered.

```
        QUERY COMMAND CONSTRUCTION SCREEN
 Data base <A> in mapset <rim_test> open.  25 records.
 The SQL select query will use the current region
 and a target clause of 'distance from 596463.15 4919041.88'

where date = 10/16/89_____________________________________________
_____________________________________________________________________
_____________________________________________________________________
_____________________________________________________________________
_____________________________________________________________________
_____________________________________________________________________
_____________________________________________________________________
_____________________________________________________________________

(Enter .show on a line to review screen layout and field names.)

  AFTER COMPLETING ALL ANSWERS, HIT <ESC> TO CONTINUE
              (OR <Ctrl-C> TO CANCEL)
```

MANAGEMENT MENU

Choice 13 from the main menu presents this menu. Each item is discussed below.

```
  s.db.rim       DATA  BASE  MANAGEMENT  MENU
Data base <A> in mapset <rim_test> open.  15025 records.

  1   Make a New Data Base in Current Mapset
  2   List Available Data Bases
  3   Remove (PERMANENTLY) Data Base from Current Mapset
  4   Recover a Data Base from a RIM ASCII File
  5   Show Screen Layout of Current Data Base
  6   Backup (UNLOAD) Data Base to RIM ASCII Format File
  7   Pack the Current Data Base
```

```
  8   Read a Site list into the Current Data Base

  0   Return to Main Menu

 0_ Your selection

 AFTER COMPLETING ALL ANSWERS, HIT <ESC> TO CONTINUE
             (OR <Ctrl-C> TO CANCEL)
```

1. Use this choice to create a new *s.db.rim* data base in the current GRASS mapset. See section below on MAKE A NEW DATA BASE.

2. List available data bases. Like 2 on MAIN MENU.

3. Delete an entire data base from the current mapset. The name of the data base and additional confirmation of the action are prompted for.

4. Choice 6 allows backup of the definition and data parts of a data base to a transportable text file. To rebuild (or build for the first time) a *s.db.rim* data base from one of these text files do the following steps:

```
   # see if the rim directory exists.
ls $LOCATION/rim/sites
   # if the directory was not found, make it.
mkdir $LOCATION/rim/sites
   # change directory to it.
cd $LOCATION/rim/sites
   # have rim build the binary data base files.
rim
RIM> input '/path/to/your/textfile'
RIM> exit
```

The data base is thus created in the current mapset. Several *s.db.rim* commands should be run to verify the integrity of the newly created data base.

5. This merely shows the screen layout of the currently open data base. It is a useful way to quickly see the layout and review the field names and types.

6. When backing up to a text file, the RIM UNLOAD command is run with the output directed to a file of the user's choice. See 4 above. It is wise to do this operation after extensive changes or additions of data records. The resulting text file can be written to tape for preservation, or shared with other GRASS systems, if desired.

7. After deleting a large number of site records, some "wasted" disk space will be present in the binary data base files. This procedure will perform an unload and a reload automatically to recover this unusable disk space. If there is any problem reopening the data base after packing, the user is notified and can recover in various ways depending on the backups which have been done.

8. Data may be loaded into a data base from an existing GRASS site_list. This procedure will prompt for the site_list name and then add the sites to the currently open data base. If all sites in the list have a comment field of the form "#value ....", the value is used as the data base site number, otherwise the sites are numbered sequentially beginning with 1. Only the site number and location coordinates are loaded for each site record by this procedure; other fields may be later added with the "change" function. See .read_sites.

MAKE A NEW DATA BASE

After entering the name of the new data base you wish to create (7 characters maximum), you then decide how to input the information required. This input may be from a text file, or may be entered directly using the editor of your choice; the former is recommended.

See .make for the way to define a data base and record (form) layout.

NOTES

This program is included in the GRASS 4.0 release, but is not automatically compiled with other GRASS commands. The user must compile this program separately.

s.db.rim interfaces to the RIM program. Both *rim* and *s.db.rim* contain FORTRAN code. The user must have access to a FORTRAN compiler in order to compile and use *s.db.rim*. See the FILES section, below, for the location of source code.

A "date" type field should be added to future versions. This version only allows storing of dates as strings (unless the user codes them to integers), and thus only string type searches can be made for dates.

BUGS

On LINUX I have some problems:

- If You select menu 13/8 and hit ESC, there is no output to the screen. *s.db.rim* expects here the name of the site file or the keyword *list*.
- If You select menu 8, You hould be asked for the site_list filename, but the screen remains empty. Simply enter the name.

FILES

The source code for RIM is located under $GISBASE/../src.related/rim

The source code for *s.db.rim* is located under $GISBASE/../src.garden/grass.rim/s.db.rim

SEE ALSO

The RIM User's Manual by Jim Fox, Academic Computing Services, Univ. of Washington. See especially Appendix B on redistribution of RIM.
The RIM Installers manual.

GRASS 4.0 Installation Guide, by Jim Westervelt and Michael Shapiro, U.S. Army Construction Engineering Research Laboratory

d.icons, d.points, g.mapsets, g.region, r.mask, s.in.ascii, s.menu, s.out.ascii, v.db.rim

AUTHORS

James Hinthorne and David Satnik, GIS Laboratory, Central Washington University, Ellensburg, WA.