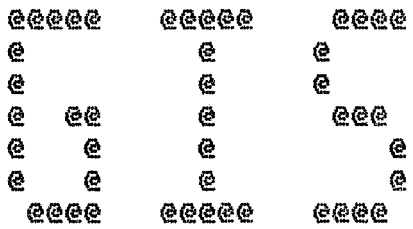


VERSION 1

# R E F E R E N C E  M A N U A L

7/29/83

by

Jim Westervelt
Michael O'Sh

This manual is a reference manual for the INGRES data base system. It documents the use of INGRES in a very terse manner. To learn how to use INGRES, refer to the document called ``A Tutorial on INGRES''.

Each entry in this manual has one or more of the following sections:

NAME section
This section repeats the name of the entry and gives an indication of its purpose.

SYNOPSIS section
This section indicates the form of the command (statement). The conventions which are used are as follows:

Bold face names are used to indicate reserved keywords.
Lower case words indicate generic types of information which must be supplied by the user; legal values for these names are described in the DESCRIPTION section.
Square brakets ( [] ) indicate that the enclosed item is optional.
Braces ( {} ) indicate an optional item which may be repeated. In some cases they indicate simple (non-repeated) grouping; the usage should be clear from context.

When these conventions are insufficient to fully specify the legal format of a command a more general form is given and the allowable subsets are specified in the DESCRIPTION section.

DESCRIPTION section
This section gives a detailed description of the entry with references to the generic names used in the SYNOPSIS section.

EXAMPLE section
This section gives one or more examples of the use of the entry.

SEE ALSO section
This section gives the names of entries in the manual which are closely related to the current entry or which are referenced in the description of the current entry.

BUGS section
This section indicates known bugs or deficiencies in the command.

ACKNOWLEDGEMENTS

We would like to acknowledge the people who have worked on INGRES in the past:

William Goran
Carl Rich
L. Van Warren
Jim Westervelt
Toyin O

NAME

DEFINITIONS - Definitions for terms used with GIS

SYNOPSIS

(Not a program - DEFINITION consists only of this text)

DESCRIPTION

Following are definitions of terms and concepts used in the Geographical Information System (GIS)

BOOLEAN - A method of logically combining information sets used as an analysis tool in GIS. Gridcell databases using the logical operators AND, OR, NOT, and GROUP are combined to create new gridcell databases. For example, the GIS tool COMBINE will produce a new gridcell database as a result to the request: "Show all the locations in my area of interest which contain (soil types 1, 3, 4, OR 10) AND (vegetation types 4 OR 5) BUT NOT slope categories 5, 6, 7, OR 8." The new gridcell database is then available for graphic display, or itself can be used in a further analysis request.

CATEGORY - Each gridcell database (layer) contains a category value for each gridcell. These numeric values (0-255) each correspond to a specific quality of the landscape. For example, a vegetation gridcell database may contain three different values (0-3), where 1 = grass, 2 = trees, and 3 = rocks. The category 0 is generally reserved to indicate no data, or data unknown.

CELL - While GIS uses gridcell and polygon data bases, nearly all analysis is done on the gridcell databases. These data bases are comprised of checkerboard style maps with category values assigned to each checkerboard square, or cell.

DIRECTORY - A computer organizes user created information into directories and files. Under the operating system UNIX, directories contain user specified collections of further directories and files. Operating system commands allow the user complete control over the creation and management of directories and files. The GIS user can choose to manage these through the GIS tools or directly interface with the powerful capabilities of the UNIX operating system.

DISTANCE-FROM - Often, GIS analyses require the calculation of distances. A distance-from analysis uses as input a grid-cell database containing the categories that a user wishes to have distances calculated from. For example, an analysis may require that the distance from each cell to the closest cell containing a permanent water body be known. A distance-from analysis will, using the user specified categories (permanent water bodies) do these calculations

and create a new gridcell database.

DOTMAP - Some of the maps produced for graphic display by GIS are actually composed of many dots being switched on or off (being printed, or not printed). Dot-matrix graphic printers work in this way. To display such a map, the computer must make an electronic dot-map with pieces of computer memory turned on and off. Dotmap refers to either the graphic display or the electronic image created or stored in memory.

FILE - A computer organizes user created information into directories and files. Files are logical collections of computer memory. Each file can contain data, text, or computer instructions (unreadable). Logical collections of are combined into computer directories.

GIS - Standing for Geographical Information System, GIS's are collections of computer capabilities used to create, manage, and analyze geographical information.

GRIDCELL - Geographical information is stored in computer memory in two forms, polygon, and gridcell. A gridcell database, layer, is simply a checkerboard representation of a map; each cell represented by a numeric category value.

LAYER - Checkerboard style gridcell databases are referred to as layers. Each is generally constructed around one theme, or logical set of categories. Most complete data bases (set of layers) have, for example, vegetation, geology, elevation, and land-use data layers.

LOCATION - Before any GIS tool can function, it must know which of the available data bases are to be used. Data bases are organized by location. Identification is accomplished easily with the GIS tool, SETUP.

MAP - Map has several meanings in the GIS context. It will generally, however, refer to graphic renditions of computer data bases containing geographical information. These may be displayed on a color or b/w monitor, printed, or plotted. Map will occasionally be used in reference to the computer data bases containing the map information.

MONITOR - Reference to a monitor implies graphics. While most GIS systems will use a monitor separate from the keyboard and screen through which the user interacts, terminals such as the (DEC) VT125 can double as a graphics monitor.

OPERATING SYSTEM - Every computer which communicates with people

manages that communication through the talents of the operating system. While the computer hardware itself is the brain, is is the operating system which provides the instructions and patterns of how to use the brain. Each operating system has its own set of user commands which, when mastered, give the user full power over the operation of the computer. The operating system under which GIS runs is called UNIX. GIS tools are actually commands to the UNIX operating system. For the convenience of the new user however, knowledge of the operating system is not initially necessary. The user who wishes to establish greater power and control over the use of the computer will wish to learn this operating system, UNIX.

POLYGON - Geographical information is stored in computer memory in two forms, polygon, and gridcell. Polygon data bases consist of sequences of coordinates which define (curved and straight) lines. These data bases are principally used to make more detailed line drawings of geographic features.

PROJECT - Before any GIS tool can function, it must know which existing, or new project is being worked on. Several projects may be going on at one time; each may run over the course of several days to many years. Users should break GIS analyses into logical projects. For example, a set of analyses on the forestry potential of a particular watershed should be a different project from the analysis of archaeological site recovery potential in another area.

Identification is accomplished easily with the GIS tool, SETUP.

SESSION - Each time a user sits down at a terminal, signs on to the computer, works, and signs off.

STATUS - Associated with each project is a status file containing tool execution information. Each time most GIS tools are used, the tool leaves a trace in a status file indicating when the tool was run, by whom, and when (and how) it finished. This information can be printed backwards (last tool used printed first) using the GIS tool status.

TOOL - This GIS is much more than an information retrieval system in that it provides a growing number of capabilities to analyze geographical data bases. Much as the carpenter uses hammer and saw to construct a house, the GIS user uses analysis tools to construct an analysis result. These tools are actually computer programs run independantly from each other. One tool creates a distance-from map; another a boolean analysis using several map layers; while another graphically displays data or results.

The more skillful the carpenter and GIS user become with their tools, the better the products of their efforts.

SEE ALSO

"ETIS Geographical Information System: USER'S MANUAL", USACERL-EN

DIAGNOSTICS

Diagnostics are written to the Location/Project's status file. See the manual entry for status.

NAME

gis - Menu of Geographic Information System Progams

SYNOPSIS

gis

DESCRIPTION

Gis (Geographic Information System) invokes the GIS menu. Intended for the new and occasional user of the ETIS - geographic information system, .I gis provides an easy and straightforward method for using the many individual analysis tools.

After typing gis, you are presented with some gis execution environment variable choices. These are displayed in a "visual-ask" question-answer session. "Visual" refers to the fact that several questions are posed at one time; you, the user, have the opportunity to answer these prompts in whatever order you choose. Only when you are satisfied with all answers do you ask the program to continue.

You are asked to identify the following:

LOCATION: This is the code name for the area you are interested in working with.

PROJECT: This is a code name (of your choice) for the particular project you (or your group) may be working with. It is permissable to have several projects going on at once. Projects carry over from one sit-down session to another over days or weeks. A project dies only at your request.

GRAPHICS MONITOR TYPE: The gis graphics commands are designed to support several graphics monitors. To get useful graphics, it is necessary to designate the type of monitor in use. In general, the system provided default answer here will be correct.

MONITOR TTY LOCATION: The electronic "location" of the graphics monitor must also be known to the system. Generally the system provided default answer here is correct also.

VERBOSE: You, the user, may choose that the menu (and some tools) operate in verbose mode. Most useful for the novice and occasional users, the verbose mode provides more useful information on the screen. This aids the user in seeing the overall structure of the menu system as well as the current location in this structure.

Your answers are automatically confined to the answer

blanks. Letters and numbers can be typed into the blanks. Several other key combinations result in different effects:

return .. This will advance the cursor to the next prompt. If the cursor was on the last prompt, hitting the return key will advance the cursor to the first prompt. Whatever was typed into the space is evaluated. The accepted answer is then left in the prompt. (Letters typed into a number prompt will be read as zeros.)

cntrl-H . (The letter "h" typed with the control key held down.) This is the common key combination to backspace the cursor.

cntrl-K . (The letter "k" typed with the control key held down.) This is the common combination to back the cursor up one line. In this case, it backs the cursor up one prompt.

cntrl-L . (The letter "l" typed with the control key held down.) This results in the entire screen being replotted. Useful in the case of a screen getting messed up with line noise.

esc ..... Once you are completely happy with all answers, hit the escape key. The program will then take and process your answers.

If your answers are within acceptable limits, you will then enter the menu proper.

Again, the menu is completely visually oriented. It provides the user with various pieces of information and several options. You will first notice that the screen has been divided into 4 boxed windows. The top box is reserved for a one-line system message. The second is used to display more complete information concerning the options. If the user has chosen verbose mode, this window displays a map of the menu, noting the current location in that map. The third window provides the current options. These options are actual gis programs and/or sub-menus. Finally, the bottom window provides the user with a place to type the next request.

Regardless of the location in the menu, the user always has several options beyond those listed:

"?" .... Typing a question mark results in more detailed information about the current options displayed on the screen.

"back" . Typing "back" will cause you to back up one

notch in the menu.

"top" .. Typing "top" will cause you to move immediately to the top of the

"status" Typing "status" will run a program which displays information about the most recent tools used in this project. This informaion is maintained over time so that you might find out what you did under the project several weeks in the past.

"exit" . "exit" will always take a user completely out of the .I gis menu.

Information at the various stages in the menu is designed to provide the user with easy signposts through the .I gis tools.

SEE ALSO

"ETIS Geographical Information System: USER"S MANUAL", USACERL-EN

DIAGNOSTICS

Diagnostics are written to the Location/Project"s status file. See the manual entry for status.

## NAME

arctogrip - conversion of a GIS arc file to a GIS grip file

## SYNOPSIS

arctogrip mapname

## DESCRIPTION

Arctogrip converts a polygon arctype file into a polygon grip file. The grip file is in a form for conversion by griptocell into a cell file.

Arctogrip searches for a file "mapname" in the subdirectories /arcs and /conv in the directory pointed to by the "LOCATION" environment variable. Using these two files for input, it generates "filename" in the subdirectory /grip.

The file in /conv identifies those areas, lines and dots defined in the /arc file to be included in the new /grip file. It takes the following form:

each record:
column 1: A for 'area', L for 'line', or D for 'dot'
column 2: Area, line or dot identification number
column 3: Category code number for this area, line, or dot

sample:

```
A    3    2
A    4    2
A    5    1
A    7    8
A    9    2
L    1   13
L    2   14
D    1   21
```

## FILES

| | | |
|---|---|---|
| /pek/gis/tmp/Afile* | - | series of temp files for area boundaries |
| /pek/gis/tmp/Dfile | - | temp file holding the Dot records |
| /pek/gis/tmp/Lfile | - | temp file holding the Line records |
| $LOCATION/poly/* | - | location of original polygon file |
| $LOCATION/grip/* | - | location of new grip file |
| $LOCATION/conv/* | - | location of area to category code inf. |

## SEE ALSO

griptocell
GRIPS User's Manual; ESRI, Redlands, California.

## DIAGNOSTICS

In the event a set of boundaries cannot be linked, the number of that area is listed along with the smallest distance between possible boundary segments.

## BUGS

For arc files with many discreet arcs, the creation of Afiles in the gis temp directory gets out of hand. Perhaps core should be used through malloc().

NAME

area_stats - retrieves statistics for cell files

SYNOPSIS

area_stats

DESCRIPTION

Area stats results in the production of a table denoting the total area in a data layer associated with each of that layers categories. Following is the result of area stats run on the cell_file "geol" for the mcclellan data base.

Map ID#: 4-8-83geol Title: geology Total Cells 156800

| Item# | Name | Acres | Hectares | Sq. mi |
|---|---|---|---|---|
| 0 | (no data) | 549.80 | 222.50 | 0.86 |
| 1 | Midway plus (A) | 433.04 | 175.25 | 0.68 |
| 2 | Floyd Shale (A) | 5215.05 | 2110.50 | 8.15 |
| 3 | Fort Payne plus (A) | 690.03 | 279.25 | 1.08 |
| 4 | Frog Mountain Sandstone (A) | 401.54 | 162.50 | 0.63 |
| 5 | Chickamauga Limestone (A) | 368.80 | 149.25 | 0.58 |
| 6 | Little Oak plus (A) | 1279.36 | 517.75 | 2.00 |
| 7 | Newala plus (A) | 1713.64 | 693.50 | 2.68 |
| 8 | Chepultepec plus (A) | 36179.15 | 14641.50 | 56.53 |
| 9 | Conasauga Formation (A) | 9356.44 | 3786.50 | 14.62 |
| 10 | Rome Formation (A) | 8722.63 | 3530.00 | 13.63 |
| 11 | Shady Dolomite (A) | 6461.66 | 2615.00 | 10.10 |
| 12 | Weisner Formation (A) | 21699.09 | 8781.50 | 33.91 |
| 13 | Talladega Slate (A) | 3792.99 | 1535.00 | 5.93 |

SEE ALSO

cell_stats(1)

NAME

cell_stats - retrieves statistics for cell files

SYNOPSIS

cell_stats

DESCRIPTION

Cell stats results in the production of a table denoting the total area in a data layer associated with each of that layers categories. Following is the result of cell stats run on the cell_file "geol" for the mcclellan data base.

Map ID#: 4-8-83geol Title: geology Total Cells: 156800

| Item# | Name | Cells | Sq. km | Percent |
|---|---|---|---|---|
| 0 | (no data) | 890 | 2.22 | 0.57 |
| 1 | Midway plus (A) | 701 | 1.75 | 0.45 |
| 2 | Floyd Shale (A) | 8442 | 21.10 | 5.38 |
| 3 | Fort Payne plus (A) | 1117 | 2.79 | 0.71 |
| 4 | Frog Mountain Sandstone (A) | 650 | 1.63 | 0.41 |
| 5 | Chickamauga Limestone (A) | 597 | 1.49 | 0.38 |
| 6 | Little Oak plus (A) | 2071 | 5.18 | 1.32 |
| 7 | Newala plus (A) | 2774 | 6.94 | 1.77 |
| 8 | Chepultepec plus (A) | 58566 | 146.41 | 37.35 |
| 9 | Conasauga Formation (A) | 15146 | 37.87 | 9.66 |
| 10 | Rome Formation (A) | 14120 | 35.30 | 9.01 |
| 11 | Shady Dolomite (A) | 10460 | 26.15 | 6.67 |
| 12 | Weisner Formation (A) | 35126 | 87.81 | 22.40 |
| 13 | Talladega Slate (A) | 6140 | 15.35 | 3.92 |

SEE ALSO

area_stat(1)

NAME

cellmod - gridcell file editor

SYNOPSIS

cellmod

DESCRIPTION

Cellmod provides the capability to inspect and/or modify data in gridcell format.

User is prompted for the filename, and the width of the cell data base. The user then has the choice of looking at or modifying the data. An area of the data base is then specified. NOTE: The North-West corner cell is at location (1,1).

FILES

/xpr/gis/data/*/cell/* :Location of cell files.

BUGS

No check is made to check the sensibility of the users responses. Unknown results will occur if the chosen window is outside the data area.

NAME

clear - erase text

SYNOPSIS

clear

DESCRIPTION

Clear attempts to clear the user's local video device (monitor or terminal) of text.

NAME

coin - tabulates the mutual occurance of 2 layer's categories

SYNOPSIS

coin

DESCRIPTION

Coin results in the production of an array of numbers denoting the mutal occurances of two sets of categories (from two map layers). Following is the result of coin run on the mcclellan data base.

---

Mask: soils
Map1: geol
Map2: site

Coincidence in number of cells
Panel #1 of 1

| site | | 0 | 1 | 2 |
|---|---|---|---|---|
| g | 0 | 23889 | 3 | 2 |
| e | 1 | 11380 | 31 | 12 |
| o | 2 | 11972 | 36 | 32 |
| l | 3 | 163 | 0 | 1 |
| | 4 | 8544 | 46 | 20 |
| | 5 | 22128 | 49 | 88 |
| | 6 | 3650 | 42 | 12 |

---

Coincidences can also be tabulated in the following units:

acres
hectares
sq. kilometers
sq. miles
percent of window
cell count

NAME

combine - boolean combination of GIS map layer categories

SYNOPSIS

combine

DESCRIPTION

Combine provides an interactive language for requesting boolean combinations of map layer categories. Map layer categories (see definitions) can be combined with one another using the operators AND, OR, and GROUP. In addition, several commands are available to provide other information useful to the current analysis.

The following definitions must be understood to use the combine command language and syntax properly:

MAP : a square grid cell map whose cells may be assigned values from 0 to 256.

MAP_EXPR : an expression that results in the creation of a new MAP.
MAP_EXPR may be one of two things:
1: MAP
2: (OPERATOR MAP_EXPR1 <MAP_EXPR2 ... MAP_EXPRN> <(directive argument l

OPERATOR COMMAND : A command which manipulates one or more MAP_EXPRs. Each operator command itself is a MAP_EXPR. There are 4 basic operators, with each being represented using one or more words or codes. Following is a list of the available operators:

| | | | |
|---|---|---|---|
| NAME OPERATOR : | "name" | | |
| AND OPERATOR : | "and" | "&" | "&&" |
| OR OPERATOR : | "or" | "\|" | "\|\|" |
| NOT OPERATOR : | "not" | "~" | |
| GROUP OPERATOR: | "group" | "grp" | "#" |

OTHER COMMANDS : Commands which are not operator commands. Following is a list of those available:

| | | | | |
|---|---|---|---|---|
| QUIT : | "quit" | "q" | "exit" | "bye" |
| STATISTICS : | "statistics" | "stats" | | |
| CATEGORIES : | "categories" | "cats" | | |
| HEAD : | "head" | | | |
| EXPRESSION : | "!" | "exp" | "expr" | "EXP" |

The correct syntax for each of these commands is described individually below.

OPERATOR COMMAND SYNTAX:

EACH OF THESE COMMANDS IS A VALID MAP_EXPR

(name map_name MAP_EXPR-A)
1) Writes MAP_EXPR-A to a file by the name "name", by which it can be recalled later.
2) Produces a MAP_EXPR which is exactly the same as MAP_EXPR-A.

(and MAP_EXPR-A MAP_EXPR-B)
( & MAP_EXPR-A MAP_EXPR-B)
( && MAP_EXPR-A MAP_EXPR-B)
ANDS MAP_EXPR-A with MAP_EXPR-B to produce a MAP_EXPR containing values of 1 and 0.

( or MAP_EXPR-A MAP_EXPR-B)
( | MAP_EXPR-A MAP_EXPR-B)
( || MAP_EXPR-A MAP_EXPR-B)
ORS MAP_EXPR-A with MAP_EXPR-B to produce a MAP_EXPR containing values of 1 and 0.

(not MAP_EXPR-A) ** NOT YET AVAILABLE **
( ~ MAP_EXPR-A) ** NOT YET AVAILABLE **
NEGATES MAP_EXPR-A to produce a MAP_EXPR containing values of 1 and 0.
The same effect of can currently be obtained via a (grp 0 MAP_EXPR-A)

(group cats_expr MAP_EXPR-A )
( grp cats_expr MAP_EXPR-A )
( # cats_expr MAP_EXPR-A )
GROUPS the categories listed in cats_expr from MAP_EXPR-A to produce a MAP_EXPR.

cats_expr is a list of categories and ranges of categories.

(window MAP_EXPR-A (point list)) ** NOT YET AVAILABLE **
clips MAP_EXPR-A against the polygon given by the point list to produce a MAP_EXPR.
the resulting MAP_EXPR will be the same as MAP_EXPR-A.

(greater_than MAP_EXPR-A MAP_EXPR-B) ** NOT YET AVAILABLE **
produces a map whose contents are TRUE wherever MAP_EXPR-A is greater than MAP_EXPR-B.

(less_than MAP_EXPR-A MAP_EXPR_B) ** NOT YET AVAILABLE **
produces a map whose contents are TRUE wherever MAP_EXPR-A is less than MAP_EXPR-B.

(equal_to MAP_EXPR-A MAP_EXPR_B) ** NOT YET AVAILABLE **
produces a map whose contents are TRUE wherever MAP_EXPR-A is equal to MAP_EXPR-B.

OTHER COMMAND SYNTAX:

(quit)
(q)
(exit)
(bye)
Terminates the current interactive combine session.

(window north south west east)
sets the current area of interest to be the area within the bounded rectangle defined by north south west east. These coordinates are array coordinates for the current data base (and not UTM coordinates).

all maps output after this statement will be with respect to this bounding area.

a default window may be placed in a users project directory.

(window)
informs user of current window

(percent_of MAP_EXPR-A MAP_EXPR-B) ** NOT YET AVAILABLE **
computes the percentage of MAP_EXPR-A contained in MAP_EXPR-A to produce a SCALAR.

(statistics MAP)
( stats MAP)
outputs the statistics of the named map.
map_name must be the name of an existing map, it may not currently be a general MAP_EXPR.

(categories MAP)
( cats MAP)
outputs the categories of the named map.
map_name must be the name of an existing map, it may not currently be a general MAP_EXPR.

(window map_name)
outputs the window of the named map.
map_name must be the name of an existing map, it may not currently be a general MAP_EXPR.

( ! number)
(exp number)
(expr number)
(EXP number)
history mechanism which is replaced by a previous request in its entirety.

EXAMPLES

Here are some simple examples demonstrating input to combine.

EXAMPLE 1:
Find all the locations which have soil categories 1 through 8, 12, 15, and 20 through 23, and save the resultis in a new file called "goodsoils".

COMBINE INPUT:

```
(name goodsoils (group 1-8 12 15 soil))
(bye)
```

EXAMPLE 2:
Find all the locations which have soil categories 1 through 8 which also contain slope categories 1 or 3, but do not have current land use (landus) categories 10, 13, or 18 through 25. Put the results in the map "for_training".

COMBINE INPUT:

```
(name for_training
    ( and
        (and
            (group 1-8 soil)
            (group 1 3 slope)
        )
        (group 0
            (group 10 13 18 25 landus)
        )
    )
)
(bye)
```

NOTE: Combine ignores linefeeds and tabs. Example two could have been written all on one line, but was broken up to show the syntax. Combine begins processing a request only after the last parenthesis is closed (making a valid MAP_EXPR).

SEE ALSO

combinate(2)

DIAGNOSTICS

Invalid requests are so designated. The current request must be then reentered.

BUGS

Combine will eventually allow requests to be entered graphically.

NAME

distance - create cellfile based on distances from categories

SYNOPSIS

distance

DESCRIPTION

Distance creates a new gridcell data layer using distance calculations based on another gridcell data layer.

The user identifies a set of categories within a gridcell file from which distances to all other cells shall be derived. Distance then results in the creation of a new map showing distances from each cell in the original data base to the closest cell containing one of the user specified categories. This process is useful for aiding analyses which need to have located all areas at least 1 mile from permanent water bodies, all areas within 5 miles of existing housing, or all areas within 500 meters of known endangered species habitats.

While calculations are good to the gridcell resolution of the database, the user is prompted to specify distance zones of interest. (e.g. "Make me a map showing zone 1 as 0 - 300 km from permanent streams, zone 2 as 300 - 600 km, and zone 4 as 600-10000 km.")

USER PROMPTS

1) You are requested to identify the existing data base from which distance-from calculations shall be based, and a name (of your choice) for the new data layer which will contain the results.
2) Next identify the categories in the existing data base from which distances shall be calculated. Here you will be given a list of the names of the existing categories. You identify those categories from which distances shall be calculated by placing a one "1" in the appropriate place. A zero "0", the default value, will indicate those categories which should be ignored.
3) Finally, identify the distance-from zones desired. You have the option of identifying up to 60 continuous zones. (Continuous in the sense that each category zone's lower value is the previous zone's upper value. The first zone always has distance 0 as its lower bound.) All distances are entered in cell units; the value of the units is the north-south and east-west size of the gridcells and is given at the top of the user prompt page.

Distance now determines which of its possible methods to run to get your requested results. It starts that process running in "background" and returns control to you. These

processes can occasionally take up to an hour or more to finish, but by being in background, you are free to do other things with the computer in the meantime.

Status can be run at anytime to check for completion of the distance from calculations. Two processes are possible at this writing, TOCELL, and FROMCELL.

FILES

New and old gridcell files with accompanying support files.

SEE ALSO

status(1), distance(2), disthead(2), tocell(2), fromcell(2) "ETIS Geographical Information System: USER'S MANUAL", USACERL-EN

DIAGNOSTICS

Diagnostics are written to the Location/Project's status file. See the manual entry for status(1).

NAME

dm_check - checks for the completion of a dot-matrix displayable GIS map

SYNOPSIS

dm_check

DESCRIPTION

Dm_check responds with a message stating whether or not a dotmap created through dotmap(1) is ready for display.

SEE ALSO

dotmap(1), ptshow(1), vtshow(1)

## NAME

dm_leg - create a dotmap legend

## SYNOPSIS

dm_leg

## DESCRIPTION

Dm leg, like dotmap, creates a picture image in computer memory. The dm leg image is a legend for the last map to be produced by dotmap. After the image is created, several display programs (each specific to a display device) can be invoked to print the image (e.g. ptshow or vtshow).

Note that a legend image should be created only after the map image has been displayed. The legend image will overwrite the map image produced by dotmap.

Dm leg is run automatically with the dotmap command. Following dotmap, ptshow will display the created map and the accompanying legend automatically.

## USER PROMPTS

None.

## FILES

/tmp/raster contains the displayable image for the dotmap and the legend.

## SEE ALSO

status(1), dotmap(1)

NAME

dotmap - creation of a dot-matrix displayable GIS map

SYNOPSIS

dotmap

DESCRIPTION

Dotmap results in the production of an image in computer memory of a set of map overlays. This image may then be displayed on printers which utilize a dot-matrix printing format.

Map overlays can be any combination of existing cell or polygon (grip or arc) map data bases (see DEFINITIONS(1)).

USER PROMPTS

1) The user is first asked to identify the limits of the window on the current data location to be mapped. See window(1). Note that the last defined window is retained as the display default.
2) Also available for modification is the desired display scale and the resolution of the dots on the users printer (or other dot oriented display device). The default values are the horizontal and vertical resolution of the normally used printer.
3) Finally the user is repetitively given the following set of options:

```
MAPPING OPTIONS for OVERLAY # %d.
A-rc map        ( arc file line map)
G-rip map       (grip file line map)
C-ell map       (cell file pattern map)
D-one           (go ahead and create map)
Q-uit           (exit without creating map)
Enter letter of option please:
```

After entering A, G, or C, the user is further prompted for the name of the file to be map drawn. A cell request will also prompt for the type of pattern to be used. "Q" stops the questions and returns the user to the menu without attempting to create a map. "D" starts the actuall map-making processes in "background" and returns the user to the menu.

Dm check(1), (dot-matrix check) or status(1) should be used to check for the completion of the map image creation in memory. After the image is created, several display programs (each specific to a display device) can be invoked to print the image (e.g. ptshow(1) or vtshow(1) ).

FILES

/tmp/raster contains the displayable image for the dotmap and the legend.

SEE ALSO

status(1), dm_leg(1), ptshow(1), vtshow(1), dm_define(2), dm_cell(2), dm_grip(2), dm_arc(2)

DIAGNOSTICS

Diagnostics are written to the Location/Project's status file. See the manual entry for status(1).

## NAME

erase - erase graphics

## SYNOPSIS

erase

## DESCRIPTION

Erase attempts to clear the user's local video device (monitor or terminal) of graphics.

## NAME

griptocell - conversion of a GIS grip file to a GIS cell file

## SYNOPSIS

griptocell filename

## DESCRIPTION

Griptocell takes the existing installation 'grip' file, filename, and creates a complete 'cell' file, filename, available for analysis. Both files are located in a location's base directory identified by the environment variable LOCATION.

The user is prompted for all necessary information, making suggestions at unsuitable responses. An understanding of the grips format will make a griptocell run obvious. Refer to the Grips User's Manual.

ESRI (Environmental Systems Research Institute), Redlands, California was the original source for the principal code used by griptocell. Three steps are reflected in the process. First, grip reads the data base producing a file of fill instructions. These instructions must then be sorted by the resident UNIX sort routine. Finally, grdpst takes these instructions and creates the final gridcell file.

## FILES

| | | |
|---|---|---|
| $LOCATION/grip/* | - | the file to be filled |
| $LOCATION/cell/* | - | where the new cell file is left |
| /xpr/gis/tmp/UNSORT | - | instructions created by grip |
| /xpr/gis/tmp/BMIF | - | the sorted UNSORT Base Map Image File |
| /xpr/gis/tmp/PRIMARY | - | where the cell file is created |
| /xpr/gis/tmp/GRIPOUT | - | Fill results |

## SEE ALSO

Grips Users Manual, ESRI, Redlands, California

## DIAGNOSTICS

As the various programs execute, diagnostics are printed at the terminal. Major problems may often be tracked down through the /xpr/gis/tmp/GRIPOUT file.

## BUGS

Ocassionally a bad grip file cannot be filled by griptocell. Some problems have been difficult to track down and requires a programmer familiar with the process. Future implementations may extract the area and centroid computations already calculated by grip.

NAME

layer_info - retrieves layer information for cell files

SYNOPSIS

layer_info

DESCRIPTION

Layer info results in the production of a table listing various information about a user chosen map layer. Following is the result of layer info run on the cell_file "geol" for the mcclellan data base.

```
+-------------------------------------------------------------------+
|  Map ID# : 4-8-83geol       Title: geology map                    |
|  Project : PERMANENT        Login of Creator: richsny             |
|-------------------------------------------------------------------|
|                                                                   |
|      Type of Map:  cell          Number of Categories: 13         |
|      Resolution:   50                                             |
|      Rows:         224                                            |
|      Columns:      700                                            |
|      Total Cells:  156800                                         |
|                                                                   |
|      Array Coordinates:                                           |
|                N:    0          S:  223                          |
|                E:  699          W:    0                          |
|      UTM Coordinates:                                             |
|                N:  3735700      S:  3724500                      |
|                E:   625500      W:   590500                      |
|                                                                   |
|      Data Source:                                                 |
|      geologic map of Calhoun Co./digitized hardcopy map           |
|                                                                   |
|                                                                   |
|      Persons Involved in Creating:                                |
|      Geographics,Westervelt,Rich                                  |
|                                                                   |
|      Comments:                                                    |
|      None                                                         |
|                                                                   |
|                                                                   |
+-------------------------------------------------------------------+
```

NAME

list - list available gridcell data layers and support files

SYNOPSIS

list

DESCRIPTION

List results in the production of a list of available grid-cell data layers and associated support files. Following is a representative output of list. "XX" under the "CELL" column notes the existence of a gridcell data layer of the name listed under "LAYER NAME". "HEAD" refers to the header support file, "STATS" to the statistics support file, and "CATS" to the category names support file.

| LAYER NAME | CELL | HEAD | STAT | CATS | |
|---|---|---|---|---|---|
| BASE | XX | XX | XX | XX | permanent |
| adminar | XX | XX | XX | XX | permanent |
| bndries | XX | XX | XX | XX | permanent |
| geol.recl | | XX | XX | XX | |

USER PROMPTS

None.

NAME

mkgisman - allows initial creation of a GIS manual entry

SYNOPSIS

mkgisman

DESCRIPTION

A shellfile useful for creating gis manual entries. You are prompted for text. At the conclusion of any section of text, a ^D will take you to the next section.

FILES

/xpr/gis/man/*

BUGS

Would be nice to invoke an editor on the manual entry being created sometime.

## NAME

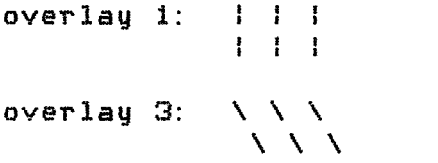
over - video graphic overlay facility

## SYNOPSIS

over

## DESCRIPTION

Over provides a means to overlay several analysis maps on a monochrome display monitor. Up to four overlays are allowed, each represented by a distinct pattern:




Three steps are accomplished by over. First, the user requested map layer is overlayed on the current map image (which may be first erased at the user's option) in computer memory. This image may be later displayed using sho over(1). Second, a statistics table is generated giving overlay coincidence information which may lster be displayed by table. Finally, the permanent map layer BASE is updated to reflect the screen image. BASE can then be used to create a hardcopy map using the tool dotmap.

## USER PROMPTS

1) The user is asked to identify the gridcell layer to be overlayed on the overlay image.
2) The display window of the data base can then be modified. (This is similar to the window modification process in window.) This window definition should be the same for any given set of overlays.
3) The overlay number, a value between 1 and 4, must be identified. The user also identifies whether the existing overlay in memory will be erased.

## FILES

/tmp/raster - file where map video image is constructed

## SEE ALSO

sho_over(1), table(1), dotmap(1), status(1)

## DIAGNOSTICS

Diagnostics are written to the Location/Project's status file. See the manual entry for status.

## BUGS

Currently works only for the VT125 terminal/monitor. Doesn't have legends, or line-drawing capabilities to allow display of coordinates or boundaries.

NAME

receive - communications interface for CROMEMCO's TERM1

SYNOPSIS

receive <filename>

DESCRIPTION

Recieve is the program which interfaces with the transfer capability of the CROMEMCO "TERM1" program. You must be in the directory where the new file is to be created. <filename> is the name of this file.

You are prompted to hit the <CLEAR> . At this point you are given the TERM1 menu by the CROMEMCO. The following menu options are useful in this process:

F - For setting the the CROMEMCO file name being accessed.

S - Send the accessed file.

T - Return to terminal mode (Talk to UNIX).

X - Exit TERM1.

Using S and F send a file. Once a file is sent, receive automatically terminates, leaving the data in <filename>.

FILES

SEE ALSO

TERM1 users manual, CROMEMCO.

DIAGNOSTICS

If more than 9 attempts to transfer a package of information fail, both receive and TERM1 stop.

BUGS

NAME

reclass - create new cell layer based on existing cell layer

SYNOPSIS

reclass

DESCRIPTION

Reclass allows a user to create a new gridcell data layer based on an existing data layer. Basically, categories in the old gridcell layer are simply converted to new categories based on the users choices.

Reclass, like all GIS programs, requires that the variables LOCATION and PROJECT be set (See setup).

Before using reclass one must know the following:

1: The category numbers associated with the old file
2: The new categories desired; and which old categories fit into which new categories.
3: The names of the new categories.

Based on the information supplied interactively by the user, the new gridcell and supporting statistics, category, and header files are created.

USER PROMPTS

1) The user is asked to identify the name of the gridcell file on which the new reclassified gridcell data shall be based, and a name for the file into which this new data shall be written.
2) Displayed to the user then is a list of the category names of the existing gridcell file. The user is prompted to enter new category numbers for each of these old categories.
3) Finally the user is prompted to enter names for these new categories.

FILES

SEE ALSO

"ETIS Geographical Information System: USER'S MANUAL", USACERL-EN

DIAGNOSTICS

Diagnostics are written to the Location/Project's status file. See the manual entry for status.

NAME

sho_over - show the current overlay image on the monitor

SYNOPSIS

sho_over

DESCRIPTION

Sho over displays the current overlay picture produced by the over tool. See the manual entry for over for more information on the construction of the image. The user chosen window is centered on the center of the screen and expanded until the north and south boundaries touch the screen top and bottom, or until the east and west boundaries touch the sides. Therefore, some images will be tall and narrow, while others will be wide and short.

If a hardcopy representation of this image is desired, the tool dotmap may be used to map the gridcell layer BASE.

Statistics of the image are printed usint the tool table.

USER PROMPTS

None

FILES

/tmp/raster - dotmap image location

SEE ALSO

over(1), table(1), status(1)

DIAGNOSTICS

Diagnostics are written to the Location/Project's status file. See the manual entry for status.

BUGS

Supports only the VT125 terminal

NAME

table - statistics table for the results of an overlay

SYNOPSIS

table

DESCRIPTION

Table prints some statistics to support the result of an overlay (see over(1)) which can be displayed with sho over(1). Over(1) provides for the overlay map layers are treated as having two categories only; 0 for nothing defined in a cell, and 1 for some category other than 0 in that cell. This process is therefore intended for display of GIS analysis results.

A sample table follows.

WINDOW STATISTICS

| COMBINATION | | CELLS | HECTARE | SQ-MILE | PERCENT |
|---|---|---|---|---|---|
| ¦ ¦ ¦ ¦ ¦ | | 1 | 0.3 | 0.0 | 0.00 |
| ¦ ¦2¦ ¦ ¦ | | 11655 | 2913.8 | 11.3 | 26.13 |
| ¦ ¦2¦3¦ ¦ | | 32599 | 8149.8 | 31.5 | 73.09 |
| ¦1¦2¦3¦ ¦ | | 345 | 86.3 | 0.3 | 0.77 |
| TOTALS: | | | | | |
| Overlay | 1 | 345 | 86.3 | 0.3 | 0.77 |
| Overlay | 2 | 44599 | 11149.8 | 43.0 | 100.00 |
| Overlay | 3 | 32944 | 8236.0 | 31.8 | 73.87 |
| Overlay | 4 | 0 | 0.0 | 0.0 | 0.00 |
| Total: | | 44600 | 11150.0 | 43.1 | 100.00 |

The first section headed "COMBINATION" is a coincident tabulation showing (in number of cells, total hectares, square miles, and percent of window) the areas which contain the different combinations of overlayed layers. The combinations are represented by the numbers between the vertical bars. "¦_¦_¦_¦_¦" represents those areas which contain none of the overlay categories. "¦_¦2¦3¦_¦" represents those areas which share overlays 2 and 3. This table can expand automatically to accomodate all 16 possible combinations of the four overlays; those containing 0 area (cells) are not printed.

The second section, "TOTALS", gives the same statistics for the individual overlays separately.

USER PROMPTS

None

FILES

$base/proj/$PROJECT/SCRCOIN

SEE ALSO

over(1), sho_over(1)

DIAGNOSTICS

Diagnostics are written to the Location/Project's status file. See the manual entry for status.

NAME

whats_here - lists category statistics for a given area

SYNOPSIS

whats_here

DESCRIPTION

Whats here results in the production of a table listing the total area associated with categories within a user chosen mask. Any cell file can be used as a mask file. All cells in that file which have a non-zero category number are "in", or "part of" the mask. All cells not assigned a category value (i.e. having a category of zero) are not part of the mask. Hence, a mask file can encompass a discrete area like a watershed, or a very diffuse area like all the cells that contain woodlands. The whats here analysis is done then by counting the occurances of the categories of up to 3 user chosen map layers within this mask. The result of a whats here run is a table listing the category names within the chosen layers and the total number of cells and percentage of the total mask area covered by that category within the mask. Below is a sample output running whats here on the mcclellan data base. "Soils" is the mask file, and "geol", "roads", and "watord" are the data layers whose categories are tabulated for occurance withing the mask.

Maskid: soils Layers: geol roads watord Total Cells: 75769

| Layer Name | Cells | % of window |
|---|---|---|
| Subcategory analysis - geol | | |
| Midway plus (A) | 5 | 0.007 |
| Floyd Shale (A) | 6107 | 8.060 |
| Fort Payne plus (A) | 844 | 1.114 |
| Frog Mountain Sandstone (A) | 493 | 0.651 |
| Chickamauga Limestone (A) | 444 | 0.586 |
| Little Oak plus (A) | 2047 | 2.702 |
| Newala plus (A) | 2502 | 3.302 |
| Chepultepec plus (A) | 26974 | 35.600 |
| Conasauga Formation (A) | 3697 | 4.879 |
| Rome Formation (A) | 2482 | 3.276 |
| Shady Dolomite (A) | 2079 | 2.744 |
| Weisner Formation (A) | 27188 | 35.883 |
| Talladega Slate (A) | 907 | 1.197 |
| Subcategory analysis - roads | | |
| (no data) | 64056 | 84.541 |
| heavy duty-4 or more lanes (L) | 73 | 0.096 |
| heavy duty-2 or 3 lanes (L) | 376 | 0.496 |

| | | |
|---|---|---|
| medium duty-2 or 3 lanes (L) | 902 | 1.190 |
| light duty-street (L) | 4368 | 5.765 |
| unimproved dirt (L) | 4314 | 5.694 |
| trail (L) | 1552 | 2.048 |
| pipeline (L) | 34 | 0.045 |
| railroad (L) | 94 | 0.124 |

| Subcategory analysis - watord | | |
|---|---|---|
| (no data) | 33 | 0.044 |
| 1st order (A) | 47186 | 62.276 |
| 2nd order (A) | 12854 | 16.965 |
| 3rd order (A) | 8474 | 11.184 |
| 4th order (A) | 5625 | 7.424 |
| 5th order (A) | 202 | 0.267 |
| 6th order (A) | 1395 | 1.841 |

NAME

window - define the bounderies of the window of interest

SYNOPSIS

window

DESCRIPTION

Window presents to the user the boundaries of the available data and the current boundaries of the rectangular window of interest. The user is allowed to change the values of the window of interest. These values are used by other GIS tools to limit the extent of an analysis or a display.

USER PROMPTS

Following is a sample prompt page that window offers. The outside boundary graphically depicts the boundary of the entire data base available. The inside boundary is that of the current chosen window. Numbers are generally listed in Universal Transverse Mercator (UTM) values. Note here that the window is the entire data base.

IDENTIFY CURRENT WORKING WINDOW

```
                                 FULL DATA BASE
 Top Row Northing:    3735700

           YOUR WINDOW
           TOP ROW NORTHING:
           3735700___

West Side  WEST SIDE                          EAST SIDE      East Side
Easting:   EASTING:                           EASTING:       Easting:
590500     590500____                         625500____      625500

           BOTTOM ROW NORTHING:
           3724500___

 Bottom Row Northing: 3724500
```

AFTER COMPLETING ALL ANSWERS, HIT <ESC> TO CONTINUE

FILES

$base/proj/$PROJECT/WIND

DIAGNOSTICS

Diagnostics are written to the Location/Project's status file. See the manual entry for status.

## NAME

dm_arc - generate dot matrix commands for creation of an arc file map

## SYNOPSIS

dm_arc -nr # -nc # -rb [ -dh # ] [ -dv # ] arcfile [ | dot-matrix ]

## DESCRIPTION

Dm_arc reads a "arc" format polygon data base and creates standard Unix-plot (see plot()) codes for the production of a linedrawing.

REQUIRED FLAGS:

- -nr Followed by the nr (number of rows) in the equivalent gridcell data base.
- -nc Followed by the nc (number of cols) in the equivalent gridcell data base.
- -rb Followed by the rb (row beginning) of the desired window. (ROWS and COLS numbered from upper left (NW)).
- -re Followed by the re (row ending) of the desired window.
- -cb Followed by the cd (col beginning) of the desired window.
- -ce Followed by the ce (col ending) of the desired window.
- -s Followed by desired scale of output window. (e.g. a 1:50,000 map would be requested by '-s 50000'.)

OPTIONAL FLAGS:

- -dv Followed by dv (dots per inch vertical) of the intended dotmatrix output device. (Default for printronix.)
- -dh Followed by dv (dots per inch horizontal) of the intended dotmatrix output device. (Default for printronix.)

OTHER:

Arcfile
must be the pathname to the desired 'arc' file.

dotmatrix'
pipes the generated Unix-plot codes into the
dotmatrix generating program
dotmatrix.
This can then be
dumped using
ptdump
or
vtdump.

Note: polymap is executed automatically with cellmap using overlay.

## FILES

/xpr/gis/data/*/poly/* location of data files

SEE ALSO

plot(1), plot(3), plot(5), ptdump, overlay
dotmatrix, cellmap, overlay

DIAGNOSTICS

Unrecognized flags terminate the process.

BUGS

dm arc does not check for inappropriate parameters:
-rb > -re beginning row greater than ending row
-cb > -ce beginning col greater that ending col
window requests outside of data area.
-nr or -nc (rows or cols) which do not agree with the data.
-s (scale) requests which would result in a map larger than 'dotmatrix' can handle.

NAME

dm_cell - generate dot matrix commands to create a cellfile map

SYNOPSIS

dm_cell -nr # -nc # -rb [ -dh # ] [ -dv # ] cellfile [ | dotmatrix ]

DESCRIPTION

Dm cell reads a "cell" format cellfile data base and creates standard Unix-plot codes for the production of a patterned drawing.

REQUIRED FLAGS:

-nr Followed by the nr (number of rows) in the equivalent gridcell data base.
-nc Followed by the nc (number of cols) in the equivalent gridcell data base.
-rb Followed by the rb (row beginning) of the desired window. (ROWS and COLS numbered from upper left (NW)).
-re Followed by the re (row ending) of the desired window.
-cb Followed by the cd (col beginning) of the desired window.
-ce Followed by the ce (col ending) of the desired window.
-s Followed by desired scale of output window. (e.g. a 1:50,000 map would be requested by '-s 50000'.)

OPTIONAL FLAGS:

-dv Followed by dv (dots per inch vertical) of the intended dotmatrix output device. (Default for printronix.)
-dh Followed by dv (dots per inch horizontal) of the intended dotmatrix output device. (Default for printronix.)

OTHER:

cellfile
is the name of the cellfile to be plotted.
dotmatrix'
pipes the generated Unix-plot codes
directly into the dotmatrix producing program.

Note: dm cell is executed automatically with dm arc using dotmap.

FILES

/xpr/gis/data/*/cell/* location of data files

SEE ALSO

plot(1), plot(3), plot(5)
dotmatrix, cellmap, overlay

DIAGNOSTICS

Unrecognized flags terminate the process.

BUGS

dm cell does not check for inappropriate parameters:
-rb > -re beginning row greater than ending row
-cb > -ce beginning col greater that ending col
window requests outside of data area.
-nr or -nc (rows or cols) which do not agree with the data.
-s (scale) requests which would result in a map larger than 'dotmatrix' can handle.

NAME

DM_DEFINE

SYNOPSIS

dm_define

DESCRIPTION

dm_define is a front end program which controls the execution of several programs designed to produce dotmatrix overlays. These include dm_arc, dm_grip, and dm_cell.

After input of all requested information, the mapping task is put into background where a 2048x2048 bitmap of the map is created. This bitmap can then be printed on a dotmatrix printer. "decmap" prints this map on a Decwriter and "ptmap" prints it onto a Printronix 600 printer.

FILES

/usr/tmp/raster /xpr/gis/data/*/cell/*
/xpr/gis/data/*/poly/*

SEE ALSO

polymap cellmap dotmatrix

DIAGNOSTICS

Unavailable installation choice is answered with a list of installations. Unavailable cell file choice is answered with a list of cell files. Unavailable polygon file choice is answered with a list of polygon files.

BUGS

Large maps mapped at scales which would produce maps larger than 2048 by 2048 dots will meet with unknown results. Overlay creates maps proportioned for the 60 dot-per-inch and 72 dot-per-inch Printronix 600. Using "polymap", "cellmap", and "dotmatrix" individually allows flexibility in the output resolution.

## NAME

dotmap - command file for easy user generation of dot matrix maps

## SYNOPSIS

dotmap

## DESCRIPTION

Dotmap is a shell file which mediates the combined execution of several processes which result in the production of a dotmatrix map displayable on the local dotmatrix printer. This process takes place with the following sequence:

The USER enters: dotmap

1 - Flag files indicating completion of map and whether map is all right are removed.
2 - dm_def (dotmatrix define) is run. This contains ALL of the user interface for construction of a map. (Instructions for the creation of the request map are placed in gis/tmp/makemap. Instructions for the creation of the accompanying legend are placed in gis/tmp/legin. If the user terminates with the desire to go ahead and create the map, the flagfile gis/tmp/makeok is created.)
3 - If the flagfile gis/tmp/makeok exists, dotmap executes the following commands in background:
  * remove the bitmap raster.
  * run dm_leg.
    Creates the legend in the bitmap.
    Print bitmap dumping instructions for using ptdump in gis/tmp/ptleg.
  * run gis/tmp/ptleg
    Turns the bitmap into Printronix print instructions.
    Results are stored in /tmp/legend.
  * remove the bitmap raster.
  * run the map creation instruction file gis/tmp/makemap.
    This calls dm_cell, dm_grip, and dm_arc as needed.
  * create the map done file gis/tmp/mapdone

## FILES

| | |
|---|---|
| /tmp/raster | storage location raw bitmap production |
| /tmp/legend | storage for the legend image |
| /gis/tmp/makemap | instructions for creation of map |
| /gis/tmp/makeok | flag denoting whether map is ok |
| /gis/tmp/mapdone | flag denoting completion of map |
| /gis/tmp/ptmap | inst. for printing section of bitmap for map |
| /gis/tmp/ptleg | inst. for printing section of bitmap for legend |

## SEE ALSO

dm_leg(2), dm_cell(2), dm_grip(2), dm_arc(2), ptdump(2), decdump(2), decshow(1), ptshow(1), status(1)

DIAGNOSTICS

Problems encountered in each of the steps are reflected through comments int he project status file; see status(1).

BUGS

NAME

dotmatrix - Unix plot style program for generating dotmatrix map images.

SYNOPSIS

```
dotmatrix < (plot command file)
(plot command creating program) | dotmatrix
```

DESCRIPTION

Dotmatrix takes standard Unix-plot commands and creates a dotmatrix file ready for printing by 'ptdump' or 'decdump'. It was adapted from the Unix code which produces Versatec plots along the lines of the Unix-plot family of commands. Plots are created in the upper left of the dotmatrix map.

FILES

/usr/tmp/raster

SEE ALSO

cellmap, polymap, overlay, plot(1), plot(3), plot(5)

DIAGNOSTICS

BUGS

If Unix-plot code attempts to create a map larger than the dot matrix can hold, unpredictable results will occur.

NAME

CREATCELL

SYNOPSIS

```
#include <gis.h>
#include <stdio.h>
creatcell(cellname, mode )          creates a cell file
     char *cellname
     int mode
```

DESCRIPTION

This is a function for interfacing with gis data gis users. (gis: Geographical Information System) Code using these functions must be compiled with the gis library; i.e. -lgis .

```
creatcell(cellname, mode)
     char *cellname
     int mode
```

Creatcell works with the Unix i/o routine create(2). The modes are equivalent to those required by these system functions. The modes for cell are the octal codes specified by chmod(1). cellname is a pointer to for example that "geol" is required instead of the complete name "/gis/data/mcclellan/cell/geol".

SEE ALSO

opencell(3), getcname_new(3)

DIAGNOSTICS

The value -1 is returned if: a needed directory is not searchable; the file does not exist, and the directory in which it is to be created is not writable; the file does exist and is unwritable; the file is a directory; there are already too many files opened.

## NAME

EXIT_MESSAGE - sends a program exit message to the project status file

## SYNOPSIS

```
exit_message(message)                    Exit message to STATUS
     char *message
```

## DESCRIPTION

Exit message can be used only after a call to gisinit(3). The message is written with the programs name (supplied by gisinit(3)) and the time and date to the current project's status file. Exit message should be used every time a program exits stating "Finished" if the process is terminating normally, or stating the problem associated with a premature exit. Aborts caused by termination signals are processed by routines established by gisinit() leaving an indication of the cause of termination in the status file automatically.

## SEE ALSO

status(), message(), gisinit()

NAME

get_date - Returns the current date and time in an ascii string

SYNOPSIS

```
#include <gis.h>
char *get_date() Returns character string with
```

DESCRIPTION

Get date returns a pointer to an ascii rendition of time and date.

## NAME

getcfile - Get a cell file with associated support files

## SYNOPSIS

```
#include <stdio.h>
getcfile_old( project, &cellfile, &headptr, &statptr, &catsptr)
getcfile_new( project, &cellfile, &headptr, &statptr, &catsptr)
getcfile ( project, &cellfile, &headptr, &statptr,
char *project
int cellfile
FILE *headptr, *statptr, *catsptr
```

## DESCRIPTION

Getcfile_old is actually a getcname_old() followed by opencstat(), openccats(), openchead(), openchist(), and openccolr(). A user is repeatedly prompted for the name of an existing map layer; the associated cell file and its support files (cell header, statistics, and category names) are returned via the argument. All files are opened for reading only. Getcfile_old then returns pointers to files associated with an existing cellfile. Getcfile_new operates similarly, but returns pointers to new, empty cellfiles. Getcfile does not discriminate between existing happily returns pointer to open files to either.

The user may exit the current program by typing "exit". Each of these calls returns with the cell file name entered.

## SEE ALSO

getcname_old(3), getcname_new(3), opencstat(3), openccats(3), openchead(3), openchist(3), openccolr(3).

NAME

limit - return a coordinate between limits

SYNOPSIS

```
limit(low, value, high)          Returns value inside low and high
     int low, value, high
```

DESCRIPTION

Limit returns low if value < low, high if value > high, and value otherwise.

NAME

make_cats - interactively generates category support files

SYNOPSIS

```
make_cats   ( cellname, number )
     char *cellname
     int number
```

DESCRIPTION

Make cats prompts the user for category names. Number indicates how many category names are needed. The results are written to the category support file for the cell data layer indicated by cellname.

DIAGNOSTICS

Make cats returns -1 if cellname cannot be opened for writing, otherwise 0 is returned.

NAME

make_chead - generates cell header/window support file

SYNOPSIS

```
make_chead ( cellname )
     char *cellname
```

DESCRIPTION

Make chead provides for the creation of an accurate cellhead/window support file for the cell file cellname.

DIAGNOSTICS

Make chead returns 0 if the cell header file already exists, or is copied from the location's cellhd/BASE file. Returns -1 if an attempt to copy failed because the cellhd file couldn't be opened.

NAME

make_colr - generates cell color support file

SYNOPSIS

```
make_colr ( cellname )
     char *cellname
```

DESCRIPTION

Make colr provides for the creation of an accurate color support file for the cell file cellname.

BUGS

NOT AVAILABLE

NAME

make_colr - interactively generates cell color support file

SYNOPSIS

```
make_hist ( cellname )
     char *cellname
```

DESCRIPTION

Make hist provides for the creation/modification of an accurate history support file for the cell file cellname.

DIAGNOSTICS

Returns -1 on failure to open a history file for reading or writing.

NAME

make_stat - generates cell statistics support file

SYNOPSIS

```
make_stat ( cellname )
     char *cellname
```

DESCRIPTION

Make stat provides for the creation of an accurate statistics support file for the cell file cellname. Make stat returns the highest category found during the count.

DIAGNOSTICS

Returns -1 on failure to open the cellfile for counting categories, or failure to open the statistics file for writing results.

## NAME

message - sends a program message to the project status file

## SYNOPSIS

```
message(message)
      char *message
```

## DESCRIPTION

Message can be used only after a call to gisinit(3). The message is written with the programs name (supplied by gisinit(3)) and the time and date to the current project's status file. Message should be used every time a program would like to leave an appropriate message in the project status file.

## SEE ALSO

status(), message(), gisinit()

## NAME

openccats - open cell category support file

## SYNOPSIS

```
#include <gis.h>
#include <stdio.h>
FILE *openccats( cellname, mode )
     char *cellname
     char *mode
```

## DESCRIPTION

Support files can be opened/created one at a time. These files are opened using fopen(3) and therefore accept modes "r" (for read), "w" (for write), "a" (for append), "r+" (for reading and writing), and "w+" (for erasing, reading, and writing). The cellname is the short name (e.g. "geol"). Returned is the pointer to the opened file.

## DIAGNOSTICS

The pointer NULL is returned if the file cannot be accessed.

## NAME

openccats - open cell color support file

## SYNOPSIS

```
#include <gis.h>
#include <stdio.h>
FILE *openccolr( cellname, mode )
     char *cellname
     char *mode
```

## DESCRIPTION

Support files can be opened/created one at a time. These files are opened using fopen(3) and therefore accept modes "r" (for read), "w" (for write), "a" (for append), "r+" (for reading and writing), and "w+" (for erasing, reading, and writing). The cellname is the short name (e.g. "geol"). Returned is the pointer to the opened file.

## DIAGNOSTICS

The pointer NULL is returned if the file cannot be accessed.

NAME

opencell - open a cell file

SYNOPSIS

```
#include <gis.h>
#include <stdio.h>
opencell( cellname, mode )          opens existing cell file
     char *cellname
     int mode
```

DESCRIPTION

Opencell works with the Unix i/o routine open(2). The modes are equivalent to those required by this system function. The modes for open are 0 (for read only), 1 (for write only), and 2 (for read and write). Cellname is a pointer to the available layer name. Note for example that "geol" is required instead of the complete name "/gis/data/mcclellan/cell/geol". Returned is the pointer to the opened cellfile.

DIAGNOSTICS

-1 is returned if the file cannot be accessed.

## NAME

openchead - open cell category support file

## SYNOPSIS

```
#include <gis.h>
#include <stdio.h>
FILE *openchead( cellname, mode )
     char *cellname
     char *mode
```

## DESCRIPTION

Support files can be opened/created one at a time. These files are opened using fopen(3) and therefore accept modes "r" (for read), "w" (for write), "a" (for append), "r+" (for reading and writing), and "w+" (for erasing, reading, and writing). The cellname is the short name (e.g. "geol"). Returned is the pointer to the opened file.

## DIAGNOSTICS

The pointer NULL is returned if the file cannot be accessed.

## NAME

openchist - open cell history support file

## SYNOPSIS

```
#include <gis.h>
#include <stdio.h>
FILE *openchist( cellname, mode )
     char *cellname
     char *mode
```

## DESCRIPTION

Support files can be opened/created one at a time. These files are opened using fopen(3) and therefore accept modes "r" (for read), "w" (for write), "a" (for append), "r+" (for reading and writing), and "w+" (for erasing, reading, and writing). The cellname is the short name (e.g. "geol"). Returned is the pointer to the opened file.

## DIAGNOSTICS

The pointer NULL is returned if the file cannot be accessed.

## NAME

opencstat - open cell statistics support file

## SYNOPSIS

```
#include <gis.h>
#include <stdio.h>
FILE *opencstat( cellname, mode )
     char *cellname
     char *mode
```

## DESCRIPTION

Support files can be opened/created one at a time. These files are opened using fopen(3) and therefore accept modes "r" (for read), "w" (for write), "a" (for append), "r+" (for reading and writing), and "w+" (for erasing, reading, and writing). The cellname is the short name (e.g. "geol"). Returned is the pointer to the opened file.

## DIAGNOSTICS

The pointer NULL is returned if the file cannot be accessed.

NAME

read_cats - read a cell file's category names

SYNOPSIS

```
#include <gis.h>
#include <stdio.h>
read_cats   (file_ptr, pcats)
     FILE *file_ptr
     struct Categories pcats
```

DESCRIPTION

Reads layer category files from file_ptr into the "Categories" structure, pcats.

NAME

read_chead - read a cell file's header/window information

SYNOPSIS

```
#include <gis.h>
#include <stdio.h>
read_chead   (file_ptr, pchead)
     FILE *file_ptr
     struct Cell_head pchead
```

DESCRIPTION

Reads layer header/window information from file_ptr into the "Cell_head" structure, pchead.

NAME

read_colr - read a cell file's color display definitions

SYNOPSIS

```
#include <gis.h>
#include <stdio.h>
read_colr   (file_ptr, pcolr)
     FILE *file_ptr
     struct Color pcolr
```

DESCRIPTION

Reads layer color display information from file_ptr into the "Color" structure, pcolr.

NAME

read_hist - read a cell file's history information

SYNOPSIS

```
#include <gis.h>
#include <stdio.h>
read_hist   (file_ptr, phist)
     struct History phist
     FILE *file_ptr
```

DESCRIPTION

Reads layer history information from file_ptr into the "History" structure, phist.

NAME

read_stat - read a cell file's statistics information

SYNOPSIS

```
#include <gis.h>
#include <stdio.h>
read_stat   (file_ptr, pstats)
     FILE *file_ptr
     struct Statistics pstats
```

DESCRIPTION

Reads layer statistics information from file_ptr into the "Statistics" structure, pstats.

NAME

VISUAL ASK LIBRARY

SYNOPSIS

```
# include <vask.h>

V_clear() ;

V_line ( line_num, text) ;
     int line_num ;
     char *text ;

V_const ( &value, type, row, col, length ) ;
V_ques  ( &value, type, row, col, length ) ;
     int value ; -OR- float value ; -OR- char *value ;
     char type ;
     int row, col, length ;

V_call () ;
```

DESCRIPTION

The visual-ask library provides an easy means to communicate with a program user one page at a time. That is, a page of text can be provided to the user with information and question prompts. The user is allowed to move the cursor from prompt to prompt answering questions in any desired order. Users answers are confined to the programmer specified locations. These functions make use of the curses library and termcap descriptions. Like vi, the user must first specify the type of terminal being used.

The standard use of this library consists of the following calls: (1) Calls to V_line() specify the text to be placed on the screen at the next call to V_call(). (2) Calls to V_const() specify placement of variables which will be displayed at the next call to V_call(). (3) Calls to V_ques() specify placement of prompt fields (and current values of a variable assigned to that field) at the next call to V_call(). (4) A call to V_call() then initiates the visual_ask library's interactive session for accumulating user inputs to the fields designated by the calls to V_ques(). (5) On return from V_call(), the program is given all answers to process. Further calls to V_call() can be made to gather information determined inappropriate. Once the currently defined screen is no longer needed (6) a call to V_clear() will clear the screen and reinitialize the questions, constants, and lines to zero.

Compilations must specify the vask, curses, and termlib libraries: -lvask -lcurses -ltermlib . These libraries must be loaded in this order. vask uses functions in curses

which uses functions in termlib.

V_clear()

V_clear() effectively clears current screen description information. Before a new (second, third, etc.) screen description, V_clear must be used to ensure that parts of the last description do not show up on the new screen.

V_line ( row, text) ;

"row" is an integer value of 0-23 specifying the row on the screen where the text is placed. The top row on the screen is row 0. "text" is a character string specifying the text for the line.

```
V_const ( &value, type, row, col, length ) ;

V_ques  ( &value, type, row, col, length ) ;
```

These two calls use the same syntax. V_ques specifies the location on the screen where a user is prompted for an entry which will be stored at the pointer specified by "type". V_const however results in the placement of a constant on the screen which the user cannot access.

"&value" is a pointer to an int, float, or char string. "type" specifies what the pointer does point to. 'i' - int, 'f' - float, and 's', character string. "row" is an integer value of 0-23 specifying the row on the screen where the value is placed. The top row on the screen is row 0. "col" is an integer value of 0-80 specifying the column on the screen where the value is placed. "length" specifies the number of columns that the value will use.

Currently, you are limited to 20 constants and 80 variables.
V_call() ;

V_call makes the call to the function which writes the desired text and constants to a static screen. It intefaces with the user, collecting information until the user is satisfied. All user determined answers are left in the desired variables.

An EXAMPLE:

```
#include <vask.h>
main()
{
     int anint ;
     float afloat ;
     char charstr[20] ;

     V_line( 5, "  Enter an Integer ") ;
     V_line( 7, "  Enter a Decimal  ") ;
     V_line( 9, "  Enter a character string ") ;

     V_ques ( &anint,   'i', 5, 30,  5) ;  /* a 5 space integer */
     V_ques ( &afloat,  'f', 7, 30,  5) ;  /* a 5 space float   */
     V_ques ( &charstr, 's', 9, 30, 10) ;  /* a 10 space string */

     V_call() ;

     printf ("%d   %f   %s0, anint, afloat, charstr) ;
}
```

At V_call, the user is presented with the screen described

in the above code. The user has several options.

<CR> moves the cursor to the next prompt window.
CNTRL-K moves the cursor to the previous prompt window.
CNTRL-H moves the cursor back one space.
Displayable ascii characters are displayed.
ESC returns control to the calling program.

The cursor cannot be moved out of the prompt windows. Before continuing to a next window, the function reads the window and rewrites what it understands. For example, text will get turned to a 0 for an integer or float, and 54.87 will be understood as 54 for an integer.

SEE ALSO
DIAGNOSTICS
BUGS

NAME

writ_cats - write a cell file's category names

SYNOPSIS

```
#include <gis.h>
#include <stdio.h>
writ_cats   (file_ptr, pcats)
     FILE *file_ptr
     struct Categories pcats
```

DESCRIPTION

Writes layer category files from the "Categories" structure pcats, into file_ptr.

NAME

writ_chead - write a cell file's header/window information

SYNOPSIS

```
#include <gis.h>
#include <stdio.h>
writ_chead   (file_ptr, pchead)
     FILE *file_ptr
     struct Cell_head pchead
```

DESCRIPTION

Writes layer header/window information from the "Cell_head" structure, pchead. into file_ptr.

NAME

writ_colr - write a cell file's color display definitions

SYNOPSIS

```
#include <gis.h>
#include <stdio.h>
writ_colr   (file_ptr, pcolr)
     FILE *file_ptr
     struct Color pcolr
```

DESCRIPTION

Writes layer color display information from the "Color" structure, pcolr into file_ptr.

NAME

writ_hist - read a cell file's history information

SYNOPSIS

```
#include <gis.h>
#include <stdio.h>
writ_hist   (file_ptr, phist)
     struct History phist
     FILE *file_ptr
```

DESCRIPTION

Writes layer history information from the "History" structure, phist into file_ptr.

NAME

writ_stat - write a cell file's statistics information

SYNOPSIS

```
#include <gis.h>
#include <stdio.h>
writ_stat   (file_ptr, pstats)
     FILE *file_ptr
     struct Statistics pstats
```

DESCRIPTION

Write layer statistics information from the "Statistics" structure, pstats file_ptr.