CONFERENCE ROOM

Center for Advanced Computation

UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN
URBANA, ILLINOIS 61801

CAC Document No. 38

IRIS/NARIS User Manual

June 30, 1972

CAC Document No. 38

IRIS/NARIS User Manual

This work supported in part
by a grant from the Ford Foundation

June 30, 1972

ACKNOWLEDGEMENT

NARIS is being developed by the Center for Advanced Computation of the University of Illinois at Urbana-Champaign in cooperation with the Northeast Illinois Natural Resource Service Center, Lisle, Illinois, and with the Northeast Illinois Planning Commission, Chicago, Illinois, and is sponsored in part by a grant from the Ford Foundation.

The NARIS computer system was designed and implemented by:

Peter A. Alsberg
Thomas S. Asbury
Calvin C. Corbin
John P. Day
Thomas P. L. Dowell
James A. Gast
William D. McTeer
Jean-Michel Michl
Stewart A. Schuster

Principal author: Calvin C. Corbin

Contributors:
Peter A. Alsberg
William D. McTeer
Stewart A. Schuster

Typist: Franklin H. Brown

TABLE OF CONTENTS

Page

Table of Contents

Page

Page

LIST OF FIGURES

LIST OF FIGURES

I. Introduction

NARIS is an acronym for Natural Resources Information System - a computerized information system developed by the Center for Advanced Computation of the University of Illinois in cooperation with the Northeast Illinois Natural Resource Service Center.

The system is designed to be used as a decision aid by public agencies and private citizens in dealing with natural resource related problems. The NARIS Database contains a wide range of information on land use as well as standardized information on geology, forestry, soil, and water characteristics for an eight county area in Northeastern Illinois.

NARIS is an information handling system. The system, itself, does not provide predictive or decision making results; it is merely a tool for making accurate and detailed information readily available to aid the planner. Thus, the responsibility for wisely using the system rests with the user.

This manual tells how to use the NARIS System. Detailed information on terminal operations and the available NARIS commands and language features is provided.

1.1 Other NARIS Publications

Other NARIS Publications and their projected audience are:

NARIS Brochure - general public

NARIS Language Guide - NARIS users

NARIS Data Guide - NARIS users

Data Coding Manual - data suppliers

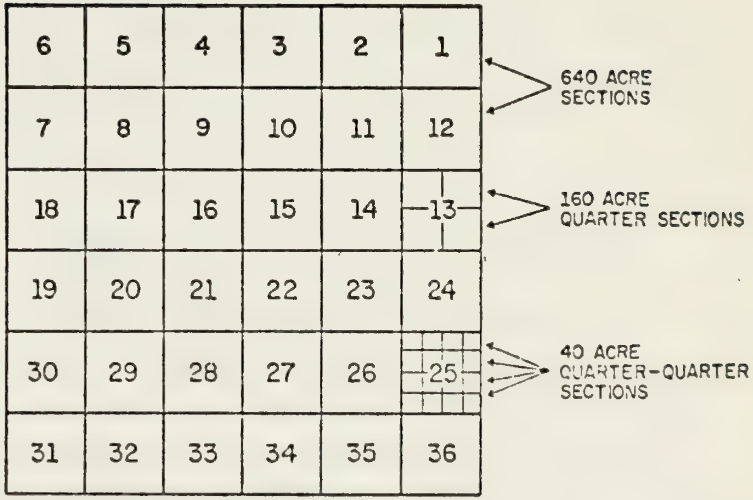
Data Input Manual - data specialists responsible for putting data into NARIS

NARIS Software Manual - computer programmers

NARIS Code Listing - computer programmers

1.3 The Database

The NARIS Database contains natural resource information pertaining to 40-acre tracts of land. Each tract is accessible by its geographic identification label which is derived from the legally established Rectangular Survey System. Each county of the State consists of surveyed townships, ranges, and sections. Sections are further subdivided into quarter sections and quarter-quarter sections. Quarter-quarter sections or equivalently, 40-acre tracts, are then used as the basic unit for storing the natural resource and socio-economic information in the NARIS Database. The figure on the following page illustrates how sections are numbered within the township; each section is further broken down into four quarter sections and each quarter section comprises four quarter-quarter sections.



6	5	4	3	2	1
7	8	9	10	11	12
18	17	16	15	14	13
19	20	21	22	23	24
30	29	28	27	26	25
31	32	33	34	35	36

Figure I-1. The township has 36 sections numbered in a winding sequence. Each section is one square mile or 640 acres and consists of four quarter sections. Each quarter section is 160 acres and consists of four quarter-quarter sections. Each quarter-quarter section is 40 acres. The 40-acre tract is the unit by which data is filed within the NARIS System.

To date, each tract (40 acres) may contain fifteen Classes of information under the following categories:

<u>Geology</u>

- Interpretation for waste disposal
- Interpretation for construction
- Water resources
- Sand and gravel resources
- Surficial deposits

<u>Land Use</u>

- HUD codes (HUD is the U.S. Department of Housing and Urban Development)
- NIPC codes (NIPC is the Northeastern Illinois Planning Commission)

<u>Forestry</u>

- Native woody vegetation
- Planted woody vegetation

<u>Soil</u>

- SCS soil characteristics (SCS is the U.S. Department of Agriculture Soil Conservation Service)

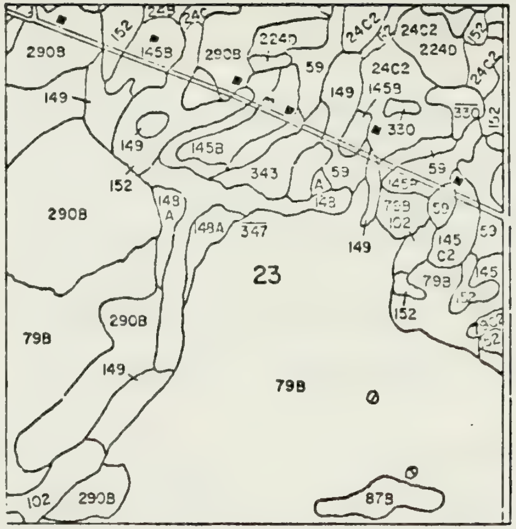
<u>Water</u>

- Watershed
- Wells
- Present impoundments
- Future impoundments
- Streams

Each Class is, in turn, made up of Data Elements. The values of these Data Elements are the attributes of each tract. For example, Soil information is collected by plots of various size within each tract. Each plot is described by the following Soil Data Elements:

- NUMBER - the Soil type of the plot;
- SLOPE - the slope of the Soil in the plot;
- EROSION - the current erosion of the Soil in the plot;
- ACRES - area of the plot in acres;
- OVERLAP - denotes whether or not the plot extends into an adjacent tract.

On the following page is a soil map for section 23 of Marengo Township in McHenry County. Within each plot is a code which can indicate the Soil NUMBER, SLOPE, and EROSION. It is from maps such as these that the Soil Data Elements are captured and then entered into the NARIS Database.



FigureI-2. Soil map for Section 23 of Marengo Township in McHenry County. Map furnished by the Soil Conservation Service.

Like Soil data, the other Classes have their own specific Data Elements. A full listing of the current Classes and the Data Elements for each Class is too long to be included here but may be found in Appendix B, NARIS Data Guide. The NARIS database is structured so that additional Classes may be easily added.

A number of agencies, coordinated by the Northeast Illinois Natural Resource Service Center in Lisle, Illinois, and the Northeastern Illinois Planning Commission, are providing the data which has been noted above. The data is coded on NARIS coding forms, keypunched, and inserted in the

Database (for details see NARIS Data Coding Manual).

1.4 The Computer System

NARIS has been programmed for a Burroughs B6700 Computer by the staff of the Center for Advanced Computation.

A language has been implemented to make use of the system as simple and straight-forward as possible, a language which tries to avoid the pitfalls of many computer languages by:

- keeping punctuation at a minimum;
- remaining English-like and allowing abbreviations which eliminate excessive repetition of lengthy phrases; and
- attempting to obtain clarification from the user whenever necessary.

II. Let's Get Started

In describing the NARIS language, some special symbols are used:

<existing region> - the left and right broken brackets are used to denote that the word(s) enclosed within them, in this case "existing region", is a generalized case of the many specific REGION names which could be used. For example,

```
FOR <existing region> TABULATE#
```

means that any name of an existing REGION or tract specification may be used as <existing region>.

{list of bad land uses} - the left and right braces are used to denote that the description within the braces is to be replaced by real values which the user has interpreted for himself. For example, the NARIS request

```
FOR <existing region> TABULATE LANDUSE1 WHERE CODE IS
ONE OF ({list of bad land uses})#
```

means that the user must choose those LANDUSE1 CODES which he considers "bad" and substitute them for "{list of bad land uses}". If one wanted to consider refuse dumps and mines (earth removal) as "bad" land uses, one might substitute the data values such that the NARIS request would be

```
FOR <existing region> TABULATE LANDUSE1 WHERE CODE IS ONE
OF (72.0, 33.0)#
```

Note that the data values, 72.0 and 33.0, are data values of LANDUSE1 CODE which mean refuse dumps and earth removal (mines), respectively - Appendix B, NARIS Data Guide.

# - This symbol is used to denote the end of a NARIS request. A carriage return does not signify the end of a NARIS request, but the end of a line on the terminal device.

Following the use of this symbol, NARIS will:

- "read" your request,
- process your request, and
- provide you with the information that had been requested.

: - The colon is the (prompt) character which NARIS will print in the first column on a line. The appearance of the colon means that NARIS is expecting input from the user. Requests should not be made of NARIS when a colon is not present - the system is processing your last request; when done, it will place a colon on the line following the last line of your output and be prepared to accept a NARIS request.

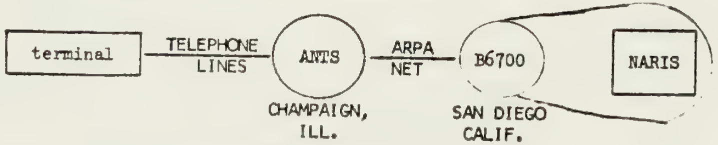
2.1 Getting In and Out of the NARIS System

Connecting the terminal to NARIS occurs in three steps:

1) Connection of the terminal to the ANTS PDP-11 system at the Center for Advanced Computation of the University of Illinois, Champaign, Illinois. ANTS is an acronym for the ARPA Network Terminal System.

2) Connection through the ANTS system over the ARPA computer network to the B6700 computer at the University of California, San Diego, California (UCSD)

3) Initiation of NARIS



2.1.1 Connecting the TI Terminal to ANTS

Set up the TI terminal. Switch settings for proper operation of the terminal are:

- the half/full switch should be at full;
- the 10/15/30 CPS switch should be at 30 CPS;
- the upper case/upper-lower case switch should be at upper case;
- the on/off switch should be at on;
- the on line key should be depressed such that the "ON LINE" light is lit.

Next, call the ANTS PDP-11 computer. The number is 217-333-7086 or

217-333-7065. If operating properly, the computer will answer with a continuous high-pitched tone. If the line is busy or you get no answer, call the ANTS computer operator at 217-333-8150 and tell him what your problem is.

When the computer answers with the tone, put the phone receiver in the cradle in the TI terminal with the cord toward the back of the terminal. Within a couple of seconds the "carrier detect" light immediately below the phone cradle should light. This indicates that the TI terminal has picked up the background frequency of the computer tone.

Now, type the character "Q" to get the computer's attention. Notice that "Q" does not appear on the paper roll. Type "Q" about once a second until "Q" appears (should not require more than 5 times). Now hold down the "control key" (CTRL) and type "P". The ANTS system should reply with the message:

```
{time of day}
A N T S  OF:  {date and time}
```

If the "Q" never prints or the "carrier detect" light goes out there is one of threee difficulties: the TI terminal is bad, the telephone connection is noisy, or the ANTS computer system is down.

All one can do to correct the TI or telephone is to check the switch settings and hang up and call the computer again - preferably on a different telephone. If still not getting a response to "Q", call the ANTS operator

(217-333-8150) to find out whether there is a reason the computer is not answering. If the computer is functioning properly, either your TI terminal or your telephone line is malfunctioning (these are infrequent occurrences).

After the ANTS OF message, the terminal is connected to the ANTS system. Any message beginning with the character "↑" will be understood to be an ANTS control message.

2.1.2 Connection through ANTS to the UCSD B6700 computer

Type

```
↑CONNECT UCSD
```

ANTS will reply

```
{time of day} ATTEMPTING CONNECTION TO UCSD
```

usually followed quickly by

```
{time of day} CONNECTION OPEN TO UCSD
```

Occasionally, however, one of the messages

```
{time of day} UCSD IS DEAD
        or
{time of day} CONNECTION ATTEMPT TO UCSD ABORTED
```

may appear or no message will appear for a long time (2 minutes) in which case one should ask the ANTS operator if he has any suggestions:

```
↑TO OPR DO YOU KNOW HOW LONG BEFORE I CAN USE UCSD?
```

Soon after the CONNECTION OPEN message, a message from UCSD will appear:

```
UCSD DATACOM ANSWERING ( version ) ENTER USERCODE, PLEASE-
```

You should enter your user code. (See NARIS User Liaison to get this information.) It will then ask:

```
AND YOUR PASSWORD
```

The password associated with your user code must be entered. If one inputs a mistaken user code or password, two more attempts may be made. Once the user code and password have been accepted, the computer may type a message:

```
FOR TODAY'S NEWS ANSWER YES-
```

Following the UCSD computer center news after a YES answer or following a NO answer the system will provide bookkeeping figures regarding your connection to UCSD.

If one should at any point after logging in desire to communicate with the UCSD computer operator, one may type

```
?TO OP  {message}
```

Messages from the operator will be of the form

```
FROM OP  {message}
```

2.1.3 Initiation of NARIS

Following the connection to UCSD, one initiates NARIS by typing "NARIS". There will be a wait, followed by the NARIS salutation:

...THIS IS NARIS. WHO ARE YOU?

You will now be using the NARIS system through two computer operating systems (ANTS in Illinois and UCSD in California). Any message beginning with the character "↑" will be interpreted as a message to the ANTS system (for example, input ↑HOSTS). Any message beginning with the character "?" will be interpreted as a message to the UCSD system (some of these are mentioned in error recovery below, section 2.1.7). All other messages will be given to NARIS.

The NARIS salutation will be followed by a line with a single ":". You should reply with your last name. Note that all communications with the NARIS system must have an end-of-request symbol,#. There will be a delay of 10-15 seconds while NARIS is initializing. Following the short delay NARIS will type:

ENTER YOUR REQUESTS

One may now engage in interactive retrieval.

2.1.4 Exiting from the NARIS System and the Computer

To exit from the NARIS System one should use the END request. It will cause your "unSAVEd" REGIONS, ABBREVIATIONS, and FUNCTIONS to be forgotten.

The system will reply with a GOOD BYE message telling you how much time and money you spent during this session.

Now type BYE to the UCSD system followed by a ↑CLOSE command to the

ANTS system and hang up the telephone.

2.1.5 Special Terminal Controls

Several special controls are available on the TI terminal to make typing easier. These are

control "H" - this control (caused by depressing the CTRL key while typing "H") will cause the previous character to be deleted. The terminal will backspace.

control "P" - this control will cause the current line to be deleted. The terminal will type "<-" and return to a new line.

BREAK - this control will someday cause NARIS to interrupt a long stream of output. At the present time it will be ignored. If it is necessary to interrupt output from NARIS, type ↑CLOSE; this will terminate NARIS and necessitate restarting at ↑CONNECT UCSD.

2.1.6 What to do when an input does not generate a reply in a reasonable time.

Receiving "no reply" is a fairly common symptom of trouble in an interactive computer system. The connection over the ARPA network is a complicated one with several possible locations for failure. One must undertake the following procedure to determine the recovery action to be taken:

If connected to UCSD when the trouble started, type ?STATUS. There

can be several replies

- no reply within one minute - the ANTS system is malfunctioning or the UCSD computer is dead or very overloaded. Typing ↑HOSTS should give a listing of the status of all the computer sites connected to the ARPA net.
    - no reply to ↑HOSTS means the ANTS system is malfunctioning. Under these circumstances, wait for about a minute, then begin again by typing "Q" as if you had just called ANTS (see section 2.1.1).
    - if the reply says UCSD IS DEAD; ask the ANTS operator if he knows when it will be running again.
    - if the reply says UCSD IS ALIVE; wait and try ↑HOSTS again. Either UCSD is overloaded (when things clear up you will get a reply to ?STATUS) or it is dying a lingering death (in which case ↑HOSTS will eventually show it is DEAD).

      If after 5 minutes UCSD IS ALIVE but no reply to ?STATUS has been received, type ↑CLOSE and start at ↑CONNECT UCSD.
- ?STATUS may also generate a reply in the form:

```
{date}                  {time}
ON AT                   {time}
ON LINE                 {minutes}
PROC. TIME              {seconds}
```

```
I/O TIME                {seconds}
JOB # IS                {number}
NARIS MCS - VERSION     {version information}
WAIT...                 {current activity}
(all times in Pacific Standard Time)
```

If the last two lines of this message (regarding NARIS) do not appear in a reasonable amount of time (2 minutes), NARIS is probably malfunctioning; type ?CANDE, then NARIS again. If this happens repeatedly call a NARIS programmer.

If the last two lines of the reply do appear, one must simply wait. If impatient, by all means try another ?STATUS.

2.1.7 Recovering from System Errors

The following is a description of the most common (but not in themselves common) errors made by the NARIS system or discovered in the use of the system. Errors which do not seem to be of these types should be brought to the attention of the NARIS staff. Only those difficulties about which the user can do something are described.

Symptom: the message

```
A NARIS SYSTEM ERROR CAUSED THE TERMINATION OF THIS
REQUEST
```

appears. NARIS will then restart.

Cure: This message is caused by some variety of NARIS programming error. The best short-term cure is avoidance. The error can be caused by a particular phrasing of a request or a reference to a particular REGION, ABBREVIATION, or FUNCTION. If this problem occurs more than once and the same REGION, ABBREVIATION, or FUNCTION is involved, do a WHAT IS REGION <region name>, ABBREVIATION <abbreviation name>, or FUNCTION <function name>. If this causes the NARIS error, do a FORGET REGION <region name>, etc. (don't worry about lost information, it is already gone). If the problem recurs call a NARIS programmer.

Symptom: the message

```
***INVALID CHARACTERS.  LINE DISCARDED***
```

is written after you send a line of input to the comouter. Frequent repetition of this message may indicate that the terminal needs to be serviced.

Cure: The system had detected characters in the input line which cannot occur in a NARIS request. Such characters are frequently caused by garbled telephone communications or defective terminals, and sometimes by hitting the wrong keys on the keyboard, or having the wrong switch settings on the terminal. One should retype the line which was discarded. There is little that can be done about noisy telephone lines except alerting the telephone company and/or calling again to get a different connection.

Symptom: the message

```
UCSD DATACOM IS NOW BACK ON LINE
```

appears.

Cure: This will normally take place when one is waiting for something to happen. It means the computer at UCSD has reinitialized. One has to reenter his user code and password and type "NARIS" again.

Symptom: one of the messages

```
{time}TR<number> <= UCSD CLOSED
```

or

```
NO DEST. DISCARDED
```

Cure: This message normally means UCSD IS DEAD. Type ↑HOSTS to be sure. Wait about 5 minutes for UCSD to become alive; if nothing happens, send a message to the ANTS operator asking if he knows how long UCSD will be dead.

Symptom: the message

```
{time}
    A N T S  OF: {date and time}
```

appers.

Cure: This message means the ANTS system has died and is now alive. The initialization procedure must be started again at ↑CONNECT UCSD.

The following flow chart is a summary of the information presented in section 2.1 and is presented so that one may quickly determine procedures to take when confronted with various situations.

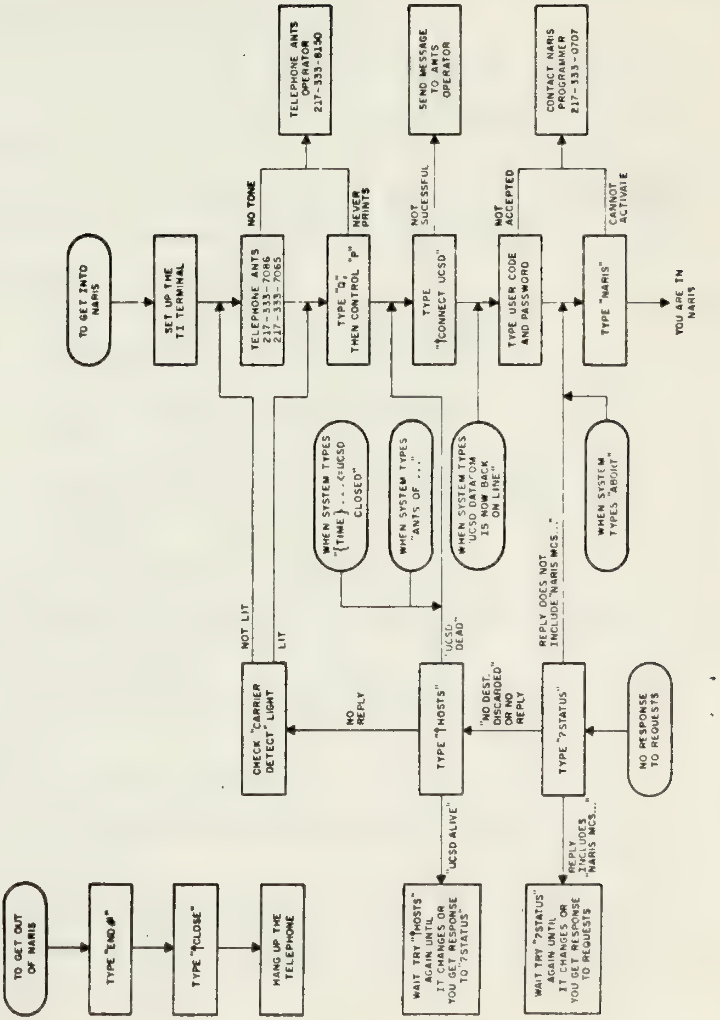

TO GET INTO NARIS
SET UP THE TI TERMINAL
TELEPHONE ANTS 217-333-7086 217-333-7065
NO TONE
TELEPHONE ANTS OPERATOR 217-333-8150
TYPE "Q;" THEN CONTROL "P"
NEVER PRINTS
TYPE "↑CONNECT UCSD"
NOT SUCESSFUL
SEND MESSAGE TO ANTS OPERATOR
TYPE USER CODE AND PASSWORD
NOT ACCEPTED
CONTACT NARIS PROGRAMMER 217-333-0707
TYPE "NARIS"
CANNOT ACTIVATE
YOU ARE IN NARIS
WHEN SYSTEM TYPES "{TIME}...<=UCSD CLOSED"
WHEN SYSTEM TYPES "ANTS OF ..."
WHEN SYSTEM TYPES "UCSD DATACOM IS NOW BACK ON LINE"
WHEN SYSTEM TYPES "ABORT"
NOT LIT
LIT
"UCSD DEAD"
REPLY DOES NOT INCLUDE "NARIS MCS..."
CHECK "CARRIER DETECT" LIGHT
NO REPLY
TYPE "↑HOSTS"
"NO DEST. DISCARDED" OR NO REPLY
TYPE "?STATUS"
NO RESPONSE TO REQUESTS
"UCSD ALIVE"
REPLY INCLUDES "NARIS MCS..."
TO GET OUT OF NARIS
TYPE "END#"
TYPE "↑CLOSE"
HANG UP THE TELEPHONE
WAIT TRY "↑HOSTS" AGAIN UNTIL IT CHANGES OR YOU GET RESPONSE TO "?STATUS"
WAIT TRY "?STATUS" AGAIN UNTIL IT CHANGES OR YOU GET RESPONSE TO REQUESTS

Diagram of the Steps Involved in Getting into or Out of NARIS

2.2 OUTPUT TO THE USER

The NARIS user may receive responses to his request for Database information

-at his terminal, or

-in the form of a computer printout listing, or

-in the form of a shaded map of a geographic area, or

-in a format of his own choosing, in which case the data would still be on the computer for further manipulation.

Output in the form of printout listings and maps are mailed from the computer site to the NARIS user.

Restructuring the data into one's own format for further analysis results in the data being saved on the computer such that the user may copy it onto a computer tape for further analysis on a computer system of his choosing.

The following diagram denotes the user in relation to his terminal, the computer, NARIS, and the different forms of output which NARIS can provide.

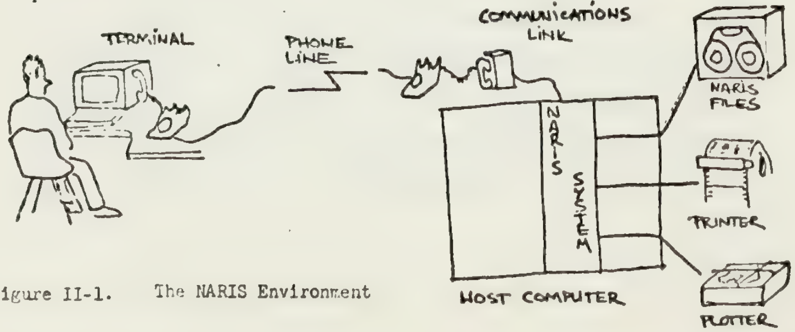


Figure II-1. The NARIS Environment

2.3 INTRODUCTORY PROBLEM USING THE LANGUAGE

Let us assume that someone would like to know what $\frac{1}{4}$ $\frac{1}{4}$ sections in Marengo Township of McHenry County are especially attractive for housing construction.

In order to use NARIS efficiently, one would need to develop a method which will produce an acceptable result before using NARIS. For example, these questions need to be answered:

·Am I familiar with the Data Classes in NARIS?

·Am I able to define data attributes which I will consider to be detrimental to housing construction?

·Do I know those data attributes which, when present in a geographic area, satisfy my requirements for housing construction?

·What kind of output do I want from NARIS?

Having answered these questions, one may order the steps necessary to produce the desired result. For example,

1) I do not want to consider any $\frac{1}{4}$ $\frac{1}{4}$ section that is subject to flooding.

2) The geographic area that I want to consider must have good geological construction characteristics and good septic and urban soil interpretations.

3) I want the land within each $\frac{1}{4}$ $\frac{1}{4}$ section to be "open" land with less than 20 acres of native, woody vegetation and having good drainage characteristics.

4) I do not want ¼ ¼ sections which contain lakes and ponds, certain types of land use, and good sand and gravel resources to be evaluated as "good" as those ¼ ¼ sections which do not have these characteristics.

Having logged into NARIS(***ENTER YOUR REQUESTS***), one can determine if there are any areas (REGIONS) in the Database which one can use by using the "LIST" request.

```
LIST REGIONS#
```

NARIS will print on your terminal all REGIONS which you may use that have already been created. MARENGOTWP is the name of one of these REGIONS. By asking,

```
WHAT IS REGION MARENGOTWP#
```

one finds that it is a REGION consisting of all ¼ ¼ sections in T44N R5E (576 ¼ ¼ sections). This is the geographic area that was stipulated in the example (Marengo Township) so it may be used. In order to exclude all ¼ ¼ sections that are subject to flooding in MARENGOTWP, one can isolate those ¼ ¼'s which do flood; e.g.,

```
REGION FLOOD IS MARENGOTWP WHERE STREAMS FLOODACRES IS NEQ O#
```

NARIS will "look" at the STREAMS FLOODACRES values in each ¼ ¼ section contained in MARENGOTWP. Those ¼ ¼'s which contain a value greater than zero (number of acres flooded) will comprise the REGION, FLOOD. REGION FLOOD contains 167 ¼ ¼ sections.

One may now construct a REGION which has not been subject to flooding by

```
REGION NOFLOOD IS MARENGOTWP EXCLUDE FLOOD#
```

In constructing REGION NOFLOOD (409 ¼ ¼ sections) NARIS will see that it is comprised of those ¼ ¼'s in MARENGOTWP which are not contained in REGION FLOOD.

From the area contained in NOFLOOD, one may create REGION SUBDIVISION such that it consists of those ¼ ¼ sections that have desirable soil interpretations and geological construction characteristics. Therefore,

```
REGION SUBDIVISION IS NOFLOOD WHERE GEOCONSTRUCT TYPE IS ONE
OF (G2,G3) AND SLIGHTSEPTICLIMIT. AND SLIGHTURBANLIMIT.#
```

Note, that a NARIS request may be several lines long. Each line must terminate with a "carriage return" to the computer. The symbol "#", of course, denotes that the request may now be processed.

REJION SUBDIVISION will consist of ¼ ¼'s which were in REGION NOFLOOD and which satisfied the conditions stipulated in the WHERE Clause:

- .data values of either "G2" or "G3" must occur for GEOCONSTRUCT TYPE - these values denote favorable geological characteristics for housing construction;
- .At least one of the SOIL conditions used to form the ABBREVIATION "SLIGHTSEPTICLIMIT." must be present; and
- .At least one of the SOIL conditions used to form ABBREVIATION "SLIGHTURBANLIMIT." must be present.

REGION SUBDIVISION (180 ¼ ¼ sections) represents those ¼ ¼'s in Marengo Township which might be good for housing construction. One might perform a number of operations on SUBDIVISION depending on specific needs.

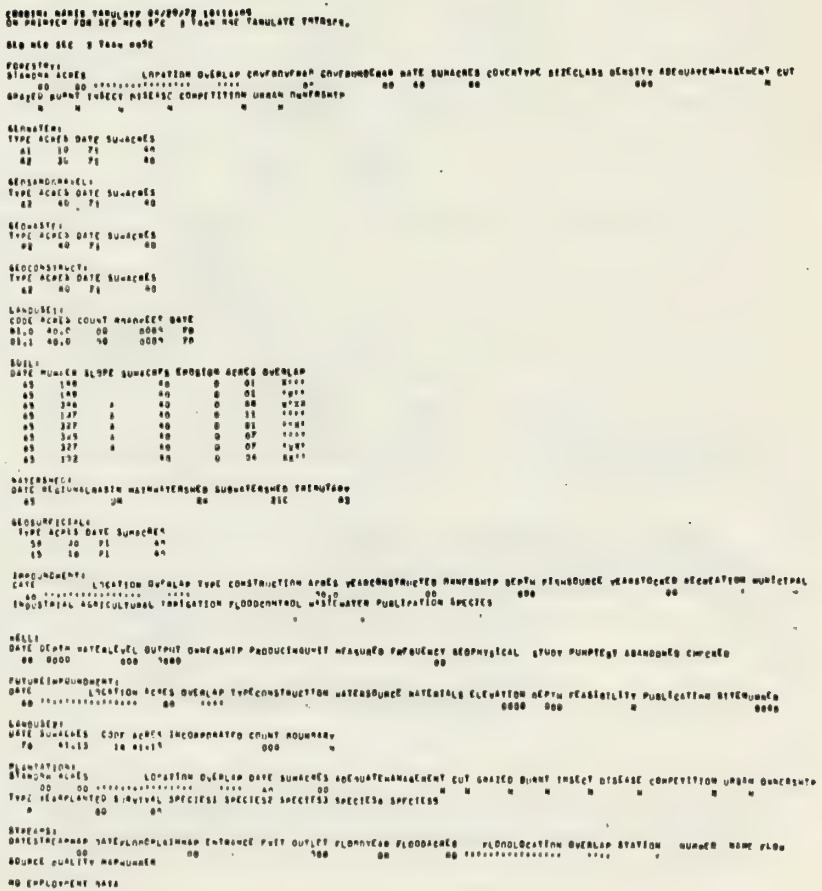
If one wanted a complete list of the data in the NARIS Database for SUBDIVISION, one could use the TABULATE request. For example,

ON PRINTER FOR SUBDIVISION TABULATE TOTRSCS.#

The output would be generated in the form of a computer print-out listing. "TOTRSCS." - total resources - is an ABBREVIATION comprising the names of all of the Data Classes in NARIS. "TABULATE TOTRSCS." results in the listing of each occurrence of each Data Class, each Data Element within an occurrenece, and the Data Element value of each Data Element for each ¼ ¼ section. The printer was selected as the means for output because of the density of information listed in each ¼ ¼ section and because 180 ¼ ¼ sections were tabulated. Figure II-2 is a computer print-out listing obtained by tabulating TOTRSCS. of one ¼ ¼ section in SUBDIVISION. Each occurrence of the Data Class is represented by a line of Data Element values; e.g., eight SOIL occurrences were recorded for the ¼ ¼ section in the figure. (See Figure II-2, TABULATE of TOTRSCS.)

One might want to determine how much farm land is present in SUBDIVISION. This could be determine by using a CALCULATE request.

FOR SUBDIVISION CALCULATE TOTAL LANDUSE1 ACRES WHERE CODE IS 81.0#



```
COMMAND: NARIS TABULATE [illegible] [illegible]
ON PRINTER FOR SEQ NEQ SEC [illegible] TABULATE TOTRSCS.

SEQ NEQ SEC [illegible]

FORESTRY:
STANDNO ACRES         LOCATION OVERLAP COVERNONFREE COVERNONWOODED DATE SUMACRES COVERTYPE SIZECLASS DENSITY ADEQUATEMANAGEMENT CUT
     00     00 ****************     ****             0*           00   40        00                         000               *
GRAZED BURNT INSECT DISEASE COMPETITION URBAN OWNERSHIP
     *     *      *       *             *     *

GEOWATER:
TYPE ACRES DATE SUMACRES
  01    10   71       40
  02    30   71       40

GEOSANDGRAVEL:
TYPE ACRES DATE SUMACRES
  02    40   71       40

GEOWASTE:
TYPE ACRES DATE SUMACRES
  02    40   71       40

GEOCONSTRUCT:
TYPE ACRES DATE SUMACRES
  02    40   71       40

LANDUSE1:
CODE ACRES COUNT ANNEXED? DATE
81.0  40.0    00     0004   70
81.1  40.0    90     0004   70

SOIL:
DATE NUMBER SLOPE SUMACRES EROSION ACRES OVERLAP
  65    198              40       0    01    [illegible]
  65    149              40       0    01    [illegible]
  65    346     A        40       0    08    [illegible]
  65    147     A        40       0    11    [illegible]
  65    327     A        40       0    01    [illegible]
  65    343     A        40       0    07    [illegible]
  65    327     A        40       0    07    [illegible]
  65    192              40       0    36    [illegible]

WATERSHED:
DATE REGIONALBASIN MAINWATERSHED SUBWATERSHED TRIBUTARY
  65            [illegible]        [illegible]     [illegible]     [illegible]

GEOSURFICIAL:
 TYPE ACRES DATE SUMACRES
   50    30   71       40
   15    10   71       40

IMPOUNDMENTS:
DATE          LOCATION OVERLAP TYPE CONSTRUCTION ACRES YEARCONSTRUCTED OWNERSHIP DEPTH FISHSOURCE YEARSTOCKED RECREATION MUNICIPAL
  00 ****************     ****                  00.0              00          000                    00           *          *
INDUSTRIAL AGRICULTURAL IRRIGATION FLOODCONTROL WASTEWATER PUBLICATION SPECIES
                                            *            *

WELLS:
DATE DEPTH WATERLEVEL OUTPUT OWNERSHIP PRODUCINGUNIT MEASURED FREQUENCY GEOPHYSICAL  STUDY PUMPTEST ABANDONED CHECKED
  00  0000        000   0000                                    00

FUTUREIMPOUNDMENT:
DATE        LOCATION ACRES OVERLAP TYPECONSTRUCTION WATERSOURCE MATERIALS ELEVATION DEPTH FEASIBILITY PUBLICATION SITENUMBER
  00 ****************    00    ****                                         0000   000           *                 0000

LANDUSE2:
DATE SUMACRES  CODE ACRES INCORPORATED COUNT BOUNDARY
  70    [illegible]    [illegible]                 000       *

PLANTATIONS:
STANDNO ACRES          LOCATION OVERLAP DATE SUMACRES ADEQUATEMANAGEMENT CUT GRAZED BURNT INSECT DISEASE COMPETITION URBAN OWNERSHIP
     00    00 ****************     ****   00       00                      *   *      *     *      *       *           *     *
TYPE YEARPLANTED SURVIVAL SPECIES1 SPECIES2 SPECIES3 SPECIES4 SPECIES5
   *          00       00

STREAMS:
DATESTREAMMAP DATEFLOODPLAINMAP ENTRANCE EXIT OUTLET FLOODYEAR FLOODACRES    FLOODLOCATION OVERLAP STATION   NUMBER  NAME FLOW
           00                00                  000        00         00 ****************    ****       *
SOURCE QUALITY MAPNUMBER

NO EMPLOYMENT DATA
```

Figure II-2. TABULATE of TOTRSCS

The NARIS response to this CALCULATE request appears on your terminal as

```
TOTAL                        OF LANDUSE1 ACRES

5785.8
```

Thus, one may note that of the 7,200 acres in SUBDIVISION, 5,787.8 acres have been recorded as "general farming or small hold cropland" (LANDUSE1 CODE 81.0).

If one wanted a graphic representation of Marengo Township with $\frac{1}{4}$ $\frac{1}{4}$ sections in REGION SUBDIVISION shaded for their "desirability according to one's preferences," one would use the MAP request. Referring to items 3) and 4) from pages II-12 and II-13, one could form the following MAP request:

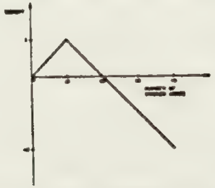
NARIS MAP Request	Interpretation
FOP SUBDIVISION MAP AT 1:40000	-Map region SUBDIVISION at a scale of 1:40,000. The following parameters determine the shading of the map:
(2* SUM LANDUSE1 ACRES WHERE CODE IS 81.0 + 2* FUNCTION [(0, 0) (10,1) (40,-2)]	-Relatively open land is desired; The FUNCTION graphed below is used to assign weight's based on wooded acres, Ten wooded acres is most desirable (greatest positive weight is assigned), If more than 20 acres, the assigned weight becomes negative; 
+ 2* SUM GEOWATER ACRES WHERE TYPE IS 01 + (-100)* SUM LANDUSE1 ACRES WHERE CODE IS ONE OF (33.0,64.0,71.0,72.0,94.0)	-Good drainage characteristics are desirable; -Mines, cemeteries, sewage disposal works, refuse dumps, and marshlands are extremely undesirable;
+ (-2)* SUM GEOSANDGRAVEL ACRES WHERE TYPE IS 01 AND ACRES GTR 10	-Tracts with sand and gravel resources are not desirable;
+ (-2)* SUM IMPOUNDMENT ACRES.	-Tracts with lakes and ponds are not desirable.

Figure II-3. A MAP Request

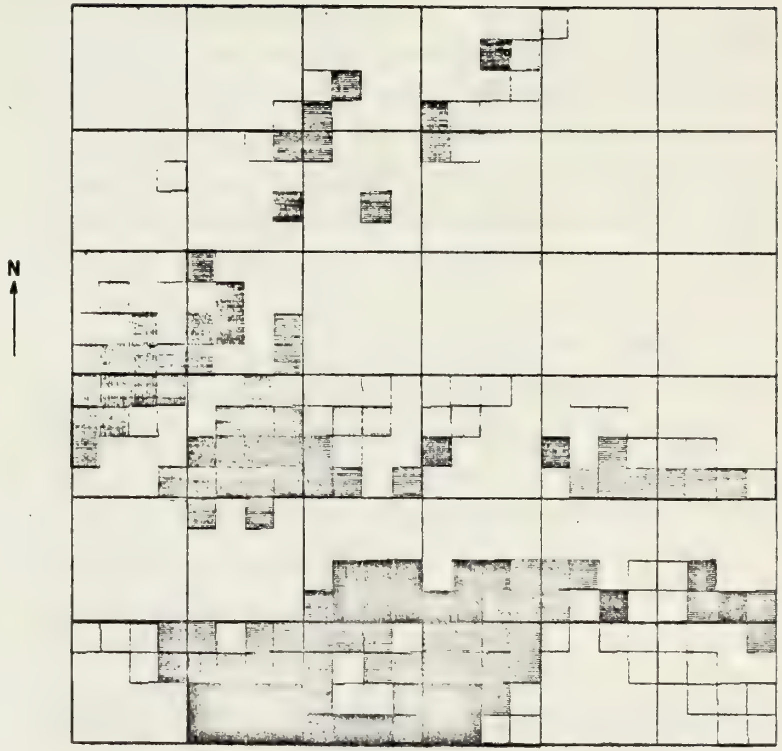


Figure II-4. This map shows which tracts are most attractive for housing construction based on the NARIS Request. The darker the shadings the higher the desirability. Only those tracts in Marengo Township are shaded which comprise REGION SUBDIVISION.

Figure II-4 is produced by this MAP request; the map would be mailed to the user.

ARITHMETIC EXPRESSIONS and a special FUNCTION were used in this MAP request in order to produce one value for each $\frac{1}{4}\frac{1}{4}$ section. The $\frac{1}{4}\frac{1}{4}$ was then mapped with shading corresponding to the "scaled" value.

Complete descriptions of the components of the language follow.

III. The NARIS Language

3.1 User Information which is Saved

The NARIS System maintains a procedure by which the user may save REGIONS, ABBREVIATIONS, and FUNCTIONS over many sessions of NARIS usage. A description of how these items may be saved is thoroughly discussed in the section, Language Constructs Which Do Not Access The Data Base (especially the constructs SAVE and FORGET). This section is devoted entirely to explaining how these items may be created.

3.1.1 REGION

The NARIS data base may only be accessed by using a REGION. A REGION may be thought of as a geographic area consisting of $\frac{1}{4}\frac{1}{4}$ sections. A REGION is, in reality, a list of $\frac{1}{4}\frac{1}{4}$ sections. A REGION may be a list containing one $\frac{1}{4}\frac{1}{4}$ section or any number of $\frac{1}{4}\frac{1}{4}$'s up to, and including, all $\frac{1}{4}\frac{1}{4}$'s in the NARIS System. <region name> may consist of any sequence of letters and digits (maximum of 35 characters) which begins with a letter and which does not correspond to a NARIS language construct.

Examples of <region names> are:

legal	not allowed
MYEXAMPLEREGIONNAME	1ACREOFG2 (number cannot be first character)
A	ILLEGALLENGTHINSPECIFYINGTHISREGIONNAME (too many characters)
SOILDATA	MARENGO TOWNSHIP (blank not allowed)

3.1.1.1 The FOR-CLAUSE-REGION

The FOR-CLAUSE-REGION may be defined as the last REGION which one used to specify the area for a TABULATE, CALCULATE, OUTPUT, or MAP request. It limits the scope of requests to a predefined area.

Using the <existing region> MARENGOTWP, one may desire to CALCULATE

the number of acres in this REGION which are plots of soil known as SOIL NUMBER 103.

```
FOR MARENGOTWP CALCULATE TOTAL SOIL ACRES WHERE NUMBER IS 103#
```

The NARIS System recognizes MARENGOTWP as a REGION and will make it a FOR-CLAUSE-REGION; prior to responding to the CALCULATE portion of the request, NARIS will print a line

```
***FOR-CLAUSE-REGION CREATED.  INCLUDES (number) TRACTS***
```

which means that the list of $\frac{1}{4}$ $\frac{1}{4}$ sections comprising the REGION is legitimate. The number which is between "INCLUDES" and "TRACTS" denotes the number of $\frac{1}{4}$ $\frac{1}{4}$'s in the REGION.

Once a FOR-CLAUSE-REGION has been created, one may submit requests without retyping

```
FOR <exisiting region>. . .#
```

NARIS will remember and use the FOR-CLAUSE-REGION which was last used in a request. For example, following the previous CALCULATE on "MARENGOTWP", one could request

```
TABULATE STREAMS WHERE FLOODACRES NEQ 0#
```

and all $\frac{1}{4}$ $\frac{1}{4}$ sections in MARENGOTWP which had a non zero data value for STREAMS FLOODACRES would be TABULATEd listing STREAMS data.

Two things must be remembered when using the FOR-CLAUSE-REGION: that is, submitting requests which will access the data base without stipulating FOR <existing region>. These are:

1) The last REGION which was used is the FOR-CLAUSE-REGION; and

2) NARIS will not remember the FOR-CLAUSE-REGION from one day to the next, therefore, one must, at least, begin the day by creating a REGION.

Some examples illustrating the use of the FOR-CLAUSE-REGION:

```
FOR REGION USER TABULATE WHERE SOIL NUMBER IS 152#

TABULATE LANDUSE1 WHERE CODE IS 33.0#
(Refers to FOR-CLAUSE-REGION  "USER")

FOR NEQ NEQ SEC 1 T44N R5E TABULATE TOTRSCS.#

CALCULATE TOTAL GEOSURFICIAL ACRES BY TYPE#
(Refers to FOR-CLAUSE-REGION "NEQ NEQ SEC 1 T44N R5E")
```

The first and third requests created REGIONS. The third request also contained a TABULATE of the total resources of the REGION.

There are three ways to create a <new region name>.

3.1.1.2 TRACT SPECIFICATION

Tract specifications may be used as a REGION and may also be used to create a <new region name>. For example,

```
FOR NEQ NEQ SEC 1 T44N R5E TABULATE TOTRSCS.#
```

will provide a listing of the total resources of the $\frac{1}{4}$ $\frac{1}{4}$ section (REGION) shown below

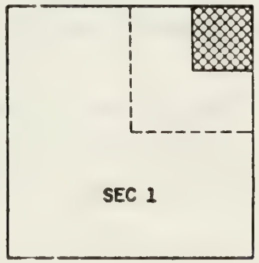


Figure III 1. NEQ NEQ SEC 1 T44N R5E

REGION <new region name> IS <tract specification>#

is a method used for creating <new region name>. One could create a REGION called ONETRACT, with the statement

REGION ONETRACT IS NEQ NEQ SEC 1 T44N R5E#

ONETRACT contains only the northeast quarter of the northeast quarter of section one in Marengo Township (T44N R5E); NEQ NEQ SEC 1 T44N R5E is the <tract specification> for this request.

The following symbols and their meanings are used in specifying a geographic area of the data base (<tract specification>) to create a REGION:

NEQ	NorthEast Quarter	SH	South Half
NWQ	NorthWest Quarter	EH	East Half
SEQ	SouthEast Quarter	NH	North Half
SWQ	SouthWest Quarter	WH	West Half

NEQ	NEQ	NorthEast Quarter of the NorthEast Quarter
NWQ	NEQ	NorthWest Quarter of the NorthEast Quarter
SEQ	NEQ	SouthEast Quarter of the NorthEast Quarter
SWQ	NEQ	SouthWest Quarter of the NorthEast Quarter
NEQ	NWQ	NorthEast Quarter of the NorthWest Quarter
NWQ	NWQ	NorthWest Quarter of the NorthWest Quarter
SEQ	NWQ	SouthEast Quarter of the NorthWest Quarter
SWQ	NWQ	SouthWest Quarter of the NorthWest Quarter
NEQ	SEQ	NorthEast Quarter of the SouthEast Quarter
NWQ	SEQ	NorthWest Quarter of the SouthEast Quarter
SEQ	SEQ	SouthEast Quarter of the SouthEast Quarter
SWQ	SEQ	SouthWest Quarter of the SouthEast Quarter
NEQ	SWQ	NorthEast Quarter of the SouthWest Quarter
NWQ	SWQ	NorthWest Quarter of the SouthWest Quarter
SEQ	SWQ	SouthEast Quarter of the SouthWest Quarter
SWQ	SWQ	SouthWest Quarter of the SouthWest Quarter

SEC or SECTION	Section 1,2,3....34,35,36
T44N	Township forty-four North
R5E	Range five east

(A complete list of the surveyed Townships in the NARIS System may be obtained from the map in Figure III-2.)

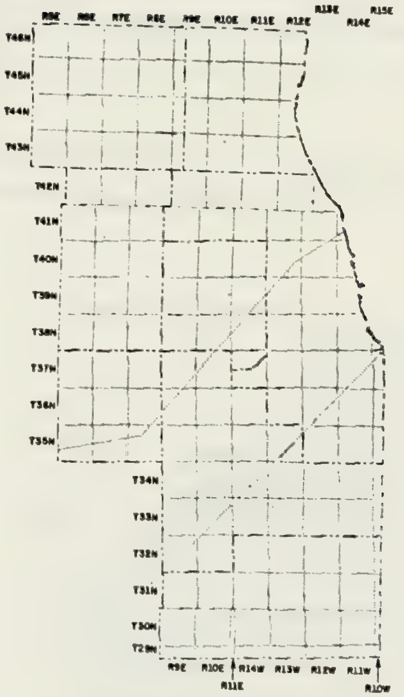


Figure III-2. Eight County map of surveyed Townships in NARIS System.

The symbols listed on the previous page may be used to specify a geographic area (REGION). For example, "SH SEC 25 T44N R5E" refers to the south half of section 25, township 44 north, range 5 east.

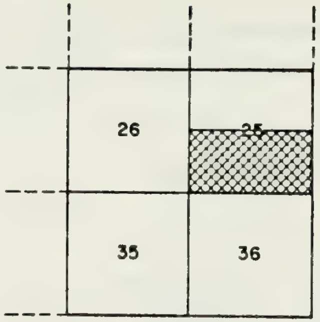


Figure III-3. SH SEC 25 T44N R5E

The NARIS notation has been expanded from that normally allowed in legal descriptions to allow one to describe blocks of survey townships with intervals and range numbers. Thus, "T41-43N R8-9E" would refer to an area comprising the six survey townships: T41N R8E, T41N R9E, T42N R8E, T42N R9E, T43N R8E, and T43N R9E.

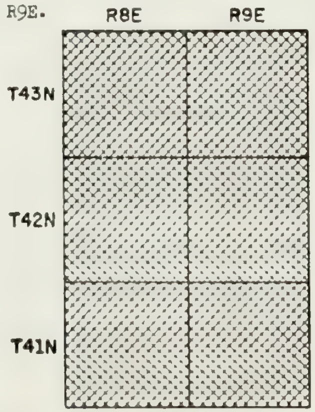


Figure III-4. T41-43N R8-9E.

Also a list of numbers or a list of numbers and intervals may be used to specify section numbers; list may be enclosed in parentheses and the numbers or intervals in the list must be separated by a comma.

The shaded area in Figure III-5 depicts the area within the survey township T36N R8E which is specified by "SEC (1,2,10-16,20-36) T36N R8E".

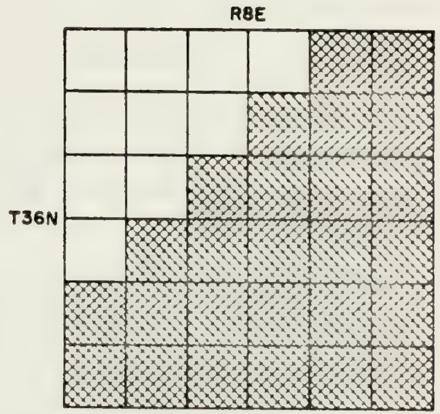


Figure III-5. SEC (1,2,10-16,20-36) T36N R8E.

In order to describe irregular subdivisions of sections, section halves and quarters may be combined. The indicated subdivisions must be enclosed in parentheses and separated by a comma.

SEC 2 T41N R8E

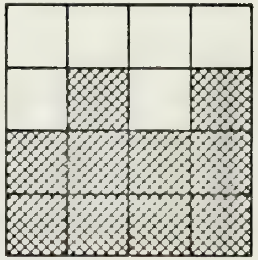


Figure III-6. (SH, SEQ NWQ, SEQ NEQ) SEC 2 T41N R8E.

Combined subsections cannot be subdivided. Specifying EH(SH, SEQ NWQ, SEQ NEQ) SEC 2 T41N R8E is not permitted as it is an attempt to subdivide combined subsections.

Some examples of creating REGIONS by using <tract specification> are:

REGION TRACTEXAMPLE1 IS SH SEC 25 T44N R5E#

REGION TRACTEXAMPLE2 IS T41-43N R8-9E#

REGION TRACTEXAMPLE3 IS SEC (1,2,10-16,20-36) T36N R8E#

REGION TRACTEXAMPLE4 IS (SH, SEQ NWQ, SEQ NEQ) SEC 2 T41N R8E#

REGION TRACTEXAMPLE5 IS NEQ SEC 1-36 T44N R5E#

The following <tract specifications> are not permitted:

NEQ T44N R5E (Section specification missing)

NEQ SEC 1 - SEC 36 T44N R5E (Incorrect section specification).
NEQ SEC 1-36 R5E T44N (Range preceeds township specification).

REGION TRACTEXAMPLE6 IS NEQ SWQ SEC 1, 2, 3, SEC 5 T44N R5E#

TRACTEXAMPLE6 comprises the northeast quarter of the southwest quarter of sections 1, 2, and 3, and all $\frac{1}{4}\frac{1}{4}$'s of section 5 while "NEQ SWQ SEC 1, 2, 3, 5 T44N R5E" would refer to only one $\frac{1}{4}\frac{1}{4}$ in section 5.

A good exercise for the reader at this point would be to "log in" to NARIS and create some REGIONS of his own by the <tract specification> method or even write out requests on a piece of paper.

3.1.1.3 REGION request with WHERE clause

Using the REGION request with a WHERE clause allows one to create REGIONS depending on their data content rather than their geographical location.

The format of creating a REGION using a WHERE clause is

REGION <new region name> IS <existing region> WHERE <where clause>#

<new region name> is the new REGION to be formed.

<existing region> is the name of an existing REGION.

WHERE is a reserved word used to preface a <where clause>.

<where clause> is a condition which must be satisfied within a $\frac{1}{4}\frac{1}{4}$ before an operation is performed. In this case, if a $\frac{1}{4}\frac{1}{4}$ in the <existing region> meets the condition stipulated in the <where clause>, the $\frac{1}{4}\frac{1}{4}$ would become a part of <new region name>. If the $\frac{1}{4}\frac{1}{4}$ does not satisfy the <where clause>, it will

not become a part of the <new region name>. A complete discussion of the WHERE clause is presented later in the manual (see Table of Contents).

It is possible to create a REGION in which each $\frac{1}{4}$ $\frac{1}{4}$ of the REGION contains FORESTRY data. Any <existing region> or <tract specification> may be used as <existing region> which contains the $\frac{1}{4}$ $\frac{1}{4}$'s to be "tested." Therefore, the following may be submitted to NARIS:

REGION FORESTRYINMARENGO IS MARENGOTWP WHERE FORESTRY ACRES NEQ 0#

NARIS will create REGION FORESTRYINMARENGO and give one the number of $\frac{1}{4}$ $\frac{1}{4}$'s which are in FORESTRYINMARENGO.

Each $\frac{1}{4}$ $\frac{1}{4}$ in FORESTRYINMARENGO will have at least one stand of native, woody vegetation (FORESTRY data).

REGIONS which are defined by a WHERE clause have special characteristics that one needs to understand in order to avoid making errors when using them. A REGION consists of a record of the data at the time the REGION was formed and will become inaccurate if the data is updated and the REGION is not. Since the rate of data revision is a relatively slow process, there is no need to feel concern about data attributes changing on a day-to-day basis. One should, however, be suspicious of very old REGIONS which were defined by using WHERE clauses.

REGIONS cannot contain partial $\frac{1}{4}$ $\frac{1}{4}$'s; therefore, when a WHERE clause is used to form a REGION, if any data attributes of the tract satisfy the condition of the WHERE clause, all of the data attributes of the tract are included in the REGION.

For example,

SEC 27 T44N R5E

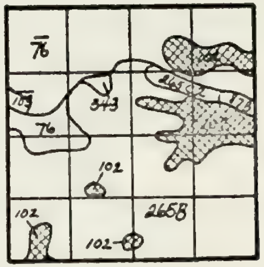


Figure III-7. Shaded area is WHERE SOIL NUMBER IS 102 (approximately 100 acres).

SEC 27 T44N R5E

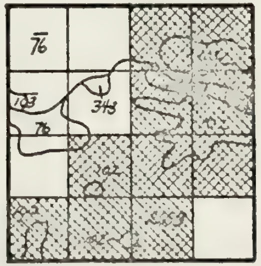


Figure III-8. Shaded area is REGIUN UNEOHTWO IS SEC 27 T44N R5E WHERE SOIL NUMBER IS 102# (approximately 400 acres)

Some examples of creating REGIONS using a WHERE clause are:

REGION WHEREEXAMPLE1 IS MARENGOTWP WHERE SOIL NUMBER IS ONE OF (103,152)#

REGION WHEREEXAMPLE2 IS MCHENRY WHERE LANDUSE2 CODE IS 18#

REGION WHEREEXAMPLE3 IS MARENGOTWP WHERE LANDUSE1 CODE IS ONE OF (33.0,91.0)#

REGION WHEREEXAMPLE4 IS FORESTRYINMARENGO WHERE STREAMS FLOODACRES EQ 0#

REGION WHEREEXAMPLE5 IS WHEREEXAMPLE3 WHERE GEOCONSTRUCT TYPE IS ONE OF (G2, G3) AND ACRES GTR 10#

REGION WHEREEXAMPLE6 IS TRACTEXAMPLE3 WHERE SOIL ACRES GTR 10#

REGION WHEREEXAMPLE7 IS SEC 2, 4, 6, 9-11 T44N R5E WHERE FUTUREIMPOUNDMENT ACRES NEQ 0#

REGION WHEREEXAMPLE8 IS T41-43N R8-9E WHERE SOIL NUMBER IS 152 AND SLOPE IS A#

Of the three methods used to create REGIONS, two have now been explained: <tract specification> and using a WHERE <where clause>. The <tract specification> method allows one to "geographically" specify any combination of $\frac{1}{4}$ $\frac{1}{4}$'s to form a REGION. The WHERE clause method allows one to specify a REGION according to the data attributes of the $\frac{1}{4}$ $\frac{1}{4}$'s. The WHERE clause method requires that an already existing REGION be used as the source of $\frac{1}{4}$ $\frac{1}{4}$'s which are to be "searched" by the WHERE <where clause>. The third method of forming a REGION allows one to form a REGION by combining two REGIONS which have previously been created.

3.1.1.4 Combination of Pre-existing REGIONS

The format of creating a REGION from pre-existing REGIONS is:

REGION <new region name> IS <existing region 1> EXCLUDE <existing region 2>#

or REGION <new region name> IS <existing region 1> INTERSECT <existing region 2>#

or REGION <new region name> IS <existing region 1> UNION <existing region 2>#

The following figures illustrate the combinations involved in the use of EXCLUDE, INTERSECT, and UNION in creating REGIONS.

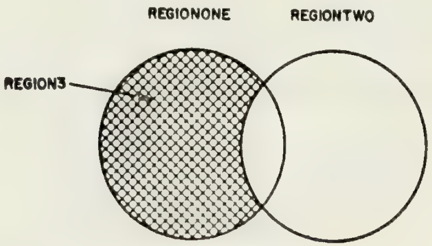


Figure III-9. REGION REGION3 IS REGIONONE EXCLUDE REGIONTWO#

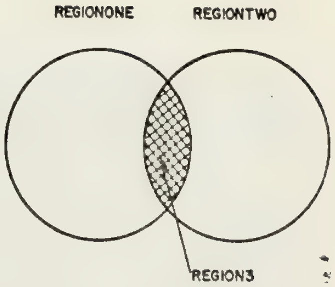


Figure III-10. REGION REGION3 IS REGIONONE INTERSECT REGIONTWO#

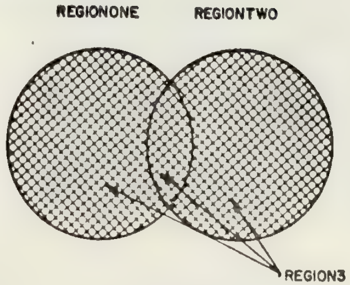


Figure III-11. REGION REGION3 IS REGIONONE UNION REGIONTWO#

EXCLUDE creates a new REGION consisting of all $\frac{1}{4}$ $\frac{1}{4}$'s in the first REGION which are not in the second REGION.

INTERSECT creates <new region name> such that it consists of those $\frac{1}{4}$ $\frac{1}{4}$'s which are common to both <existing region 1> and <existing region 2>.

UNION creates <new region name> such that all $\frac{1}{4}$ $\frac{1}{4}$'s in <existing region 1> and <existing region 2> are included in it.

Using EXCLUDE, one might create a REGION which excludes certain characteristics. For example, if one wanted to form areas within Marengo township whose tracts have not been flooded, it is possible to create REGION NOFLOOD by EXCLUDing FLOOD:

```
REGION FLOOD IS MARENGOTWP WHERE STREAMS FLOODACRES GTR 0#
```

(This request produces FLOOD, a REGION consisting of $\frac{1}{4}$ $\frac{1}{4}$'s which have had flooded acreage.)

```
REGION NOFLOOD IS MARENGOTWP EXCLUDE FLOOD#
```

(Only those tracts in Marengo township which have no recorded flood acreage -FLOODACRES equal 0- will comprise NOFLOOD.)

An example of using INTERSECT can be shown by determining the relationship between flooded $\frac{1}{4}$ $\frac{1}{4}$'s in Marengo and $\frac{1}{4}$ $\frac{1}{4}$'s with native, woody vegetation.

```
REGION WOODS IS MARENGOTWP WHERE FORESTRY ACRES GTR 0#
```

(All $\frac{1}{4}$ $\frac{1}{4}$'s in Marengo containing stands of forest will be a part of REGION WOODS.)

```
REGION WOODFLOOD IS WOODS INTERSECT FLOOD#
```

($\frac{1}{4}$ $\frac{1}{4}$'s which have data attributes denoting flood acreage and forestry stands will comprise WOODFLOOD.)

Some examples illustrating the correct language used in forming a REGION from pre-existing REGIONS are:

```
REGION PREEXAMPLE1 IS TRACTEXAMPLE1 INTERSECT WHEREEXAMPLE1#

REGION PREEXAMPLE2 IS FLOOD EXCLUDE SEC 23 T44N R5E#

REGION PREEXAMPLE3 IS SEC 23 T44N R5E UNION NEQ SWQ SEC 10 T45N
R6E#

REGION PREEXAMPLE4 IS SUBDIVISION INTERSECT WOODS#

REGION PREEXAMPLE5 IS SUBDIVISION EXCLUDE WOODS#

REGION PREEXAMPLE6 IS TRACTEXAMPLE7 UNION WHEREEXAMPLE5#
```

Figures III-12 through III-16 illustrate graphically the formation of REGION RESULT by creating REGIONS from <tract specifications> and from pre-existing REGIONS.

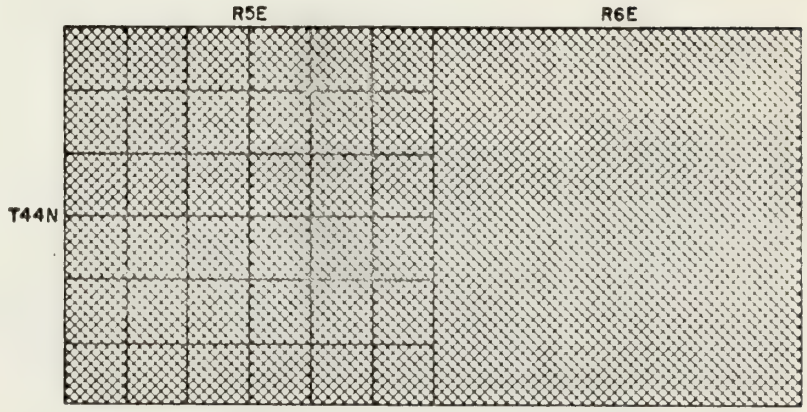


Figure III-12. REGION FIRST IS T44N R5-6E#

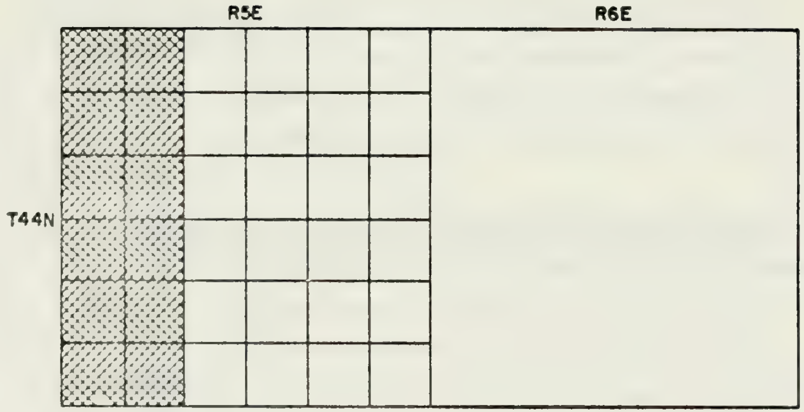


Figure III-13. REGION SECOND IS SEC(5-8, 17-20, 29-32) T44N R5E#

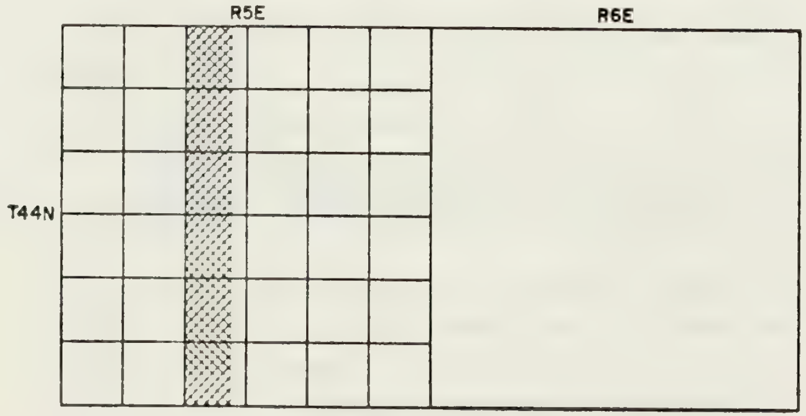


Figure III-14. REGION THIRD IS (WH EH, WH) SEC (4,9,16,21,28,33) T44N R5E#

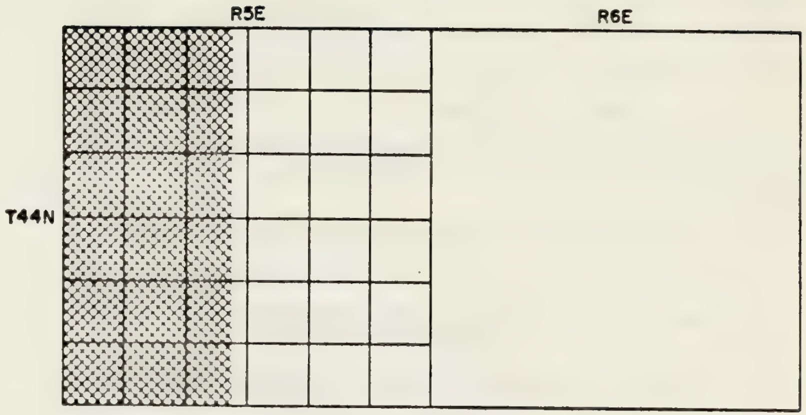


Figure III-15. REGION FOURTH IS SECOND UNION THIRD#

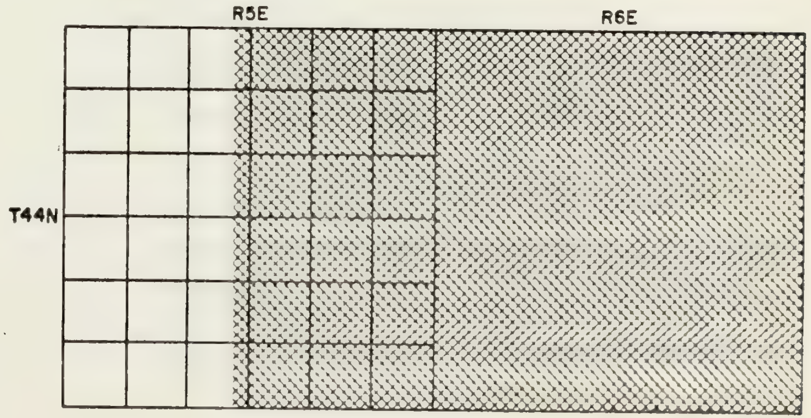


Figure III-16. REGION RESULT IS FIRST EXCLUDE FOURTH#

3.1.2 ABBREVIATIONS

An ABBREVIATION is a word which ends with a period and is created as a substitute for a phrase to be used in requests. For example, it is often desirable to do a TABULATE of the total data resources for a selected area. One may do this by

```
FOR SELECTEDAREA TABULATE GEOSURFICIAL, GEOCONSTRUCT, GEOWASTE,
GEOWATER, GEOSANDGRAVEL, SOIL, WATERSHED, LANDUSE1, LANDUSE2,
WELL, STREAMS, IMPOUNDMENT, FUTUREIMPOUNDMENT, FORESTRY, PLANTATION#
```

The series of Data Class names is very long. It would be much better if one could substitute a name for the series of classes. One may create an ABBREVIATION which means the same thing. The general definition of creating an ABBREVIATION is very much the same as creating a REGION:

```
ABBREVIATION <abbreviation name>. IS (any sequence or combination
of words, digits, blanks, and special characters)#
```

Thus, it is possible (and useful) to create an ABBREVIATION, TOTRSCS. (total data resources), which comprises all of the Data Class names as they would be used in requests (separated by commas).

```
ABBREVIATION TOTRSCS. IS GEOSURFICIAL, GEOCONSTRUCT, GEOWASTE,
GEOWATER, GECSANDGRAVEL, SOIL, WATERSHED, LANDUSE1, LANDUSE2,
WELL, STREAMS, IMPOUNDMENT, FURTUREIMPOUNDMENT, FORESTRY,
PLANTATION#
```

It should be noted that when creating an ABBREVIATION, the number of characters which follow the reserved word, IS, should not exceed 1,200 characters (about 20 full lines).

One could now perform a TABULATE on SELECTEDAREA by using TOTRSCS. by

```
FOR SELECTEDAREA TABULATE TOTRSCS.#
```

Suppose one desired to perform an operation on a number of REGIONS depending upon the content (data attributes) of the $\frac{1}{4}$ $\frac{1}{4}$'s. Let us assume that the SOIL NUMBERs used to form ABBREVIATION SILTLOAMSOILS. are in fact indications that the plots of soil are silty loam.

```
ABBREVIATION SILTLOAMSOILS. IS 27, 59, 91, 145, 146, 147, 148,
149, 155, 158, 189, 192, 193, 206, 221, 223, 224, 228, 229, 231,
241#
```

This ABBREVIATION could now be <u>used</u> in searching REGIONS for the specified SOIL NUMBERs

```
FOR SELECTEDAREA1 CALCULATE TOTAL SOIL ACRES BY NUMBER WHERE
NUMBER IS ONE OF (SILTLOAMSOILS.)#

FOR SELECTEDAREA1 TABULATE FORESTRY WHERE SOIL NUMBER IS ONE OF
(SILTLOAMSOILS.)#
```

and, to get a listing of the total resources -

```
FOR SELECTEDAREA1 WHERE SOIL NUMBER IS ONE OF (SILTLOAMSOILS.)
TABULATE TOTRSCS.#
```

Some examples of forming ABBREVIATIONS are:

```
ABBREVIATION ABBREXAMPLE1. IS CALCULATE TOTAL SOIL ACRES BY SOIL
NUMBER WHERE NUMBER IS ONE OF (SILTLOAMSOILS.)#

ABBREVIATION ABBREXAMPLE2. IS TABULATE TOTRSCS.#

ABBREVIATION ABBREXAMPLE3. IS WHERE FORESTRY ACRES GTR 10#

ABBREVIATION ABBREXAMPLE4. IS SH SEC 25 T44N R5E#

ABBREVIATION ABBREXAMPLE5. IS MARENGOTWP WHERE WELL DEPTH GTR 0#

ABBREVIATION ABBREXAMPLE6. IS SOIL NUMBER WHERE NUMBER IS ONE OF
(SILTLOAMSOILS.)#

ABBREVIATION ABBREXAMPLE7. IS ABBREXAMPLE6. AND FORESTRY ACRES
GTR 5#
```

ABBREVIATION ABBREXAMPLE8. IS FOR MARENGOTWP CALCULATE TOTAL
LANDUSE1 ACRES BY CODE#

The ABBREVIATIONS which have been created in the examples could be used in the following requests:

FOR SUBDIVISION ABBREXAMPLE1.#

FOR ABBREXAMPLE4. ABBREXAMPLE2.#

FOR TRACTEXAMPLE5 ABBREXAMPLE3. TABULATE TOTRSCS.#

REGION WELLINMARENGO IS ABBREXAMPLE5.#

FOR MARENGOTWP CALCULATE TOTAL SOIL ACRES BY ABBREXAMPLE6.#

FOR SEC 10-20 T44N R5E TABULATE ABBREXAMPLE7.#

ABBREXAMPLE8.#

It is perfectly valid to create an ABBREVIATION without placing a period after the <abbreviation name>; however, since a period is acceptable when forming an <abbreviation name>, it is used in this manual so that the reader will get into the habit of always using the period. The period following an ABBREVIATION, e.g., "<abbreviation name>.", is absolutely necessary when the ABBREVIATION is used (see the examples above).

As in WHERE clauses, ABBREVIATIONS must be created and used with care such that the NARIS interpretation of one's request coincides with the user's interpretation of his request.

3.1.3 FUNCTIONS

A thorough description of FUNCTIONS and their use is presented in

the section on Arithmetic Expressions (3.3). The creation of a <function name> request is dealt with here.

The general format of the FUNCTION <function name> request is

FUNCTION <function name> IS(<decimal number>, <decimal number>)#

The arrow pointing from the end of the closed parenthesis to in front of the open parenthesis is used to show that the use of coordinates in defining a <function name> is non-finite; thus any number (a minimum of two) of "(<decimal number>, <decimal number>)" coordinates may appear in the definition of a FUNCTION <function name>.

The coordinates enclosed in parentheses define points on an x, y axis such that

FUNCTION FUNEXAMPLE IS (-3,-2) (-1.5,0) (1-,1) (1,0) (6,3)#

would represent

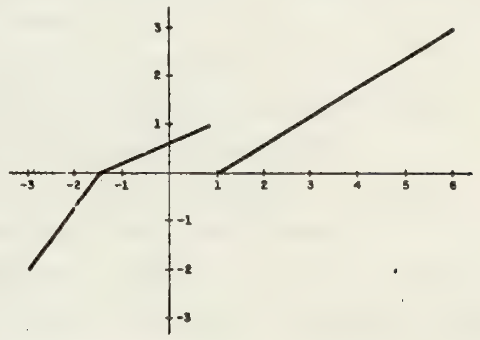


Figure III-17. x,y axis.

Note that a minus sign or a plus sign following a decimal number indicates

that number is slightly less ("-") or slightly more ("+") than the number. These signs are used to describe discontinuous functions. Some examples of creating FUNCTIONS are:

```
FUNCTION FUNEXAMPLE1 IS (0,0) (1,1) (4,-2) #
FUNCTION FUNEXAMPLE2 (0,1) (2,2) #
```

3.1.4 PUBLIC/PRIVATE

Each REGION, ABBREVIATION, and FUNCTION which a user has created is considered by the NARIS System to be PRIVATE unless the user specifies it to be PUBLIC or SEMIPUBLIC.

PUBLIC REGIONS, ABBREVIATIONS, and FUNCTIONS may be accessed and used by any user of the NARIS System.

SEMIPUBLIC REGIONS, ABBREVIATIONS, and FUNCTIONS may be accessed and used only by users in the same "user group" as the person who specified that the item be SEMIPUBLIC.

PRIVATE ABBREVIATIONS, FUNCTIONS, and REGIONS may be used only by the person who created the item.

Every REGION, ABBREVIATION, and FUNCTION is initially PRIVATE-an individual creates the item and the NARIS System notes that it "belongs" only to that person. One may however make the item(s) SEMIPUBLIC or PUBLIC if it is felt that the item(s) would be of use to other users.

See 3.2, Language Constructs which Do Not Access the Database - particularly the use of MAKE.

3.2 Language Constructs which Do Not Access the Data Base

3.2.1 SAVE

SAVE is a NARIS language construct which is used to instruct the NARIS System to retain an ABBREVIATION, FUNCTION, or REGION. The SAVEd ABBREVIATION, FUNCTION, or REGION is considered PRIVATE (rather than PUBLIC or SEMIPUBLIC) and will be accessible to the user every time he "logs in" to NARIS.

The format of SAVE is

```
SAVE ABBREVIATION <abbreviation name># or
SAVE FUNCTION <function name># or
SAVE REGION <region name>#
```

Also a series may be SAVEd, e.g.,

```
SAVE ABBREVIATION <abbreviation name>, REGION <region name>,
<region name 1>, <region name 2>, ABBREVIATION <abbreviation
name 1>, (etc.)#
```

In this case, NARIS would respond with:

```
***ABBREVIATION <abbreviation name> WILL BE SAVED***
***REGION <region name> WILL BE SAVED***
***REGION <region name 1> WILL BE SAVED***
***REGION <region name 2> WILL BE SAVED***
***ABBREVIATION <abbreviation name 1> WILL BE SAVED***
```

It should be noted that one may SAVE an ABBREVIATION, REGION, or FUNCTION any time during the session in which it was created. Also, when one "gets out of " NARIS he will have the opportunity to SAVE whatever

REGIONS, ABBREVIATIONS, and FUNCTIONS he desires. For example, let us assume that during a session on NARIS, one created the following:

- REGIONS named FIRST, SECOND, and THIRD
- ABBREVIATIONS named CALCABBR., and TABABBR.
- FUNCTION named TIMBERCOST

when one does

```
END#
```

NARIS will respond with

```
REGIONS WHICH WILL NOT BE SAVED:
FIRST
SECOND
THIRD
FOR-CLAUSE-REGION

ABBREVIATIONS WHICH WILL NOT BE SAVED:
CALCABBR
TABABBR

FUNTIONS WHICH WILL NOT BE SAVED:
TIMBERCOST
ARE THERE ANY OF THESE YOU WANT TO SAVE UNTIL NEXT TIME?
```

and NARIS will place a prompt character (:), colon, on a new line expecting an answer to its question.

If one answers with a "NO#", NARIS will FORGET (see 3.2.3) all of the REGIONS, ABBREVIATIONS, and FUNCTIONS which were listed. This means that they would not be available to the user the next time he logged into NARIS; he would have to re-create them. Following "NO#", NARIS will proceed to close the user out of the NARIS System and return him to the host computer system.

If one answers with a

```
YES#
```

NARIS will ask

```
WHICH ONES?
```

and the user is required to list those things which he wants to SAVE, e.g.,

```
SAVE REGION FIRST, THIRD, ABBREVIATION CALCABBR., FUNCTION
TIMBERCOST#
```

NARIS will reply with

```
***REGION FIRST WILL BE SAVED***

***REGION THIRD WILL BE SAVED***

***ABBREVIATION CALCABBR WILL BE SAVED***

***FUNCTION TIMBERCOST WILL BE SAVED***

ARE THERE MORE?
```

Again one may answer with a "YES#" or a "NO#". If "YES#" , NARIS will ask "WHICH ONES?" and expect to be given more names to SAVE. If "NO#", NARIS will FORGET the items which have not been SAVEd. For example,

```
***REGION SECOND FORGOTTEN***

***REGION FOR-CLAUSE-REGION FORGOTTEN***

***ABBREVIATION TABABBR FORGOTTEN***
```

and return the user to the host computer system. If should be noted that one may respond to

```
ARE THERE ANY OF THESE YOU WANT TO SAVE UNTIL NEXT TIME?
```

with neither a "YES#" nor a "NO#" but with

```
SAVE REGION FIRST, THIRD, ABBREVIATION CALCABBR., FUNCTION
TIMBERCOST#
```

It should be noted, also, that if, during a session on NARIS, one has created no new ABBREVIATIONS, REGIONS, or FUNCTIONS or has SAVEd everything NARIS will not ask if you want anything SAVEd

Some examples of using the SAVE language construct are presented here:

```
1.  REGION SAVEXAMPLE1 IS SH SEC 31 T44N R5E#

    SAVE REGION SAVEXAMPLE1#

2.  REGION SAVEXAMPLE2 IS SAVEXAMPLE1 WHERE SOIL NUMBER IS ONE
    OF (WA,103)#

    SAVE REGION SAVEXAMPLE2#

3.  ABBREVIATION SAVEXAMPLE3. IS WHERE IMPOUNDMENT ACRES GTR O#

    ABBREVIATION SAVEXAMPLETHREE. IS CALCULATE TOTAL
    FUTUREIMPOUNDMENT ACRES/40*100#

    SAVE ABBREVIATION SAVEXAMPLE3., SAVEXAMPLETHREE.#
```

Having decided to SAVE an ABBREVIATION, REGION, or FUNCTION, one should decide whether or not it should be made available to other users. NARIS views ABBREVIATIONS, REGIONS, and FUNCTIONS as belonging to a user or a group of users: PRIVATE, SEMIPUBLIC, or PUBLIC (see 3.1.4). When an ABBREVIATION, REGION or FUNCTION is created, it is consdiered PRIVATE and may be used only by the person who created it. To make the REGION, ABBREVIATION, or FUNCTION available to other users, one needs to use the NARIS language construct, MAKE.

3.2.2 MAKE

Having formed a REGION, ABBREVIATION, or FUNCTION, one may MAKE it SEMIPUBLIC or PUBLIC (note that the REGION, ABBREVIATION, or FUNCTION

which has been made PUBLIC or SEMIPUBLIC may then be made PRIVATE) by the following format:

```
MAKE PUBLIC REGION <region name># or
MAKE PUBLIC ABBREVIATION <abbreviation name># or
MAKE PUBLIC FUNCTION <function name># or
MAKE SEMIPUBLIC REGION <region name># or
MAKE SEMIPUBLIC ABBREVIATION <abbreviation name># or
MAKE SEMIPUBLIC FUNCTION <function name># or
MAKE PRIVATE REGION <region name># or
MAKE PRIVATE ABBREVIATION <abbreviation name># or
MAKE PRIVATE FUNCTION <function name>#
```

As in SAVE (3.2.1) and FORGET (3.2.3), a series or list of names may be included in one MAKE statement. The word PUBLIC, SEMIPUBLIC or PRIVATE may appear only once in a MAKE statement. For example, the following is not allowed

```
MAKE PUBLIC REGION SAVEXAMPLE1, SEMIPUBLIC REGION SAVEXAMPLE2#
```

One could, however, request

```
MAKE PUBLIC REGIONS SAVEXAMPLE1, SAVEXAMPLE2, ABBREVIATION
SAVEXAMPLE3#
```

It is also not allowed to MAKE a name PUBLIC or SEMIPUBLIC when the same name has been made PUBLIC or SEMIPUBLIC by another user. For example, if MARENGOTWP is a PUBLIC REGION, one may create one's own (PRIVATE) REGION named MARENGOTWP, however, this name cannot then be made PUBLIC as the name already exists as a PUBLIC REGION.

To make a REGION, FUNCTION, or ABBREVIATION "PRIVATE", of course,

the item must be PUBLIC or SEMIPUBLIC and belong to you (that is, you must be the person who created it and made it PUBLIC or SEMIPUBLIC). For example, the following sequence of NARIS statements MAKEs a REGION PUBLIC and then PRIVATE

```
REGION MYMAKE IS MARENGOTWP EXCLUDE SH SEC 29 T44N R5E#
MAKE PUBLIC REGION MYMAKE#
MAKE PRIVATE REGION MYMAKE#
```

It is important to be aware of what actions are taken by NARIS in regard to the statements above.

In the first statement, REGION MYMAKE is created from an <existing region> and from a <tract specification>. MYMAKE is considered by NARIS to be a "PRIVATE" REGION belonging to the person who created it. At this point, MYMAKE will not be SAVEd since NARIS has not been instructed to SAVE it.

The second statement instructs NARIS to MAKE the REGION "PUBLIC". Having been made PUBLIC, REGION MYMAKE will no longer exist as a PRIVATE REGION--accessible only to the person who created it; instead, it will be accessible to all users of the NARIS System and will automatically be SAVEd by NARIS.

The third statement (MAKE PRIVATE REGION MYMAKE#) removes REGION MYMAKE from the "PUBLIC" domain. This statement is only valid when initiated by the user who originally created MYMAKE. Any other user attempting to MAKE the PUBLIC REGION MYMAKE a PRIVATE REGION would receive an error message:

```
ERROR 236 * YOU DO NOT HAVE AUTHORITY OVER REGION MYMAKE
```

It must be noted that even though the user who created the REGION has made it PRIVATE, the REGION will not be SAVEd unless one performs a

```
SAVE REGION MYMAKE#
```

The following NARIS statement is not valid and would result in the error message noted above:

```
MAKE PRIVATE REGION MARENGOTWP#
```

"MARENGOTWP" is a PUBLIC REGION which was not created by the person who is attempting to MAKE it PRIVATE. Anyone desiring to MAKE PRIVATE MARENGOTWP may do so simply by creating their own REGION (which is by definition PRIVATE to them). For example,

```
REGION MYMARENGOTWP IS MARENGOTWP#
```

or one may even create his own REGION named MARENGOTWP (see 3.1.1).

Some examples of the use of the NARIS language construct, MAKE, are:

1. ABBREVIATION MAKEXAMPLE1. IS SOIL NUMBER WHERE NUMBER IS W#

   MAKE SEMIPUBLIC ABBREVIATION MAKEXAMPLE.#
   (As with PUBLIC, SEMIPUBLIC items are automatically SAVEd by NARIS.)

2. MAKE PUBLIC FUNCTION TIMBERCOST, ECOLOGYWEIGHT#
   (Here, TIMBERCOST and ECOLOGYWEIGHT are, of course, FUNCTION names.)

3. ABBREVIATION MYMAKEXAMPLE1. IS MAKEXAMPLE1.#

   SAVE ABBREVIATION MYMAKEXAMPLE1.#
   (These two statements produce a PRIVATE "version" of MAKEXAMPLE1.-a SEMIPUBLIC ABBREVIATION-which has been renamed and SAVEd.)

4. REGION FORESTRYINMARENGO IS MARENGOTWP WHERE FORESTRY ACRES GTR 0#

   MAKE PUBLIC REGION FORESTRYINMARENGO#

Having discussed the SAVing and MAKing of ABBREVIATIONS, REGIONS, and FUNCTIONS, a language construct which is used to remove these items is presented.

3.2.3 FORGET

The NARIS language construct, FORGET, is used to remove ABBREVIATIONS, REGIONS, and FUNCTIONS from the NARIS System.

The format of the FORGET construct is

```
FORGET REGION <region name>#

          or

FORGET ABBREVIATION<abbreviation name>#

          or

FORGET FUNCTION <function name>#
```

Again (as in SAVE 3.2.1), a series or list of names may be used, e.g.,

```
FORGET ABBREVIATION <abbreviation name>, REGION <region name>,
<region name 1>, <region name 2>#
```

NARIS would respond with

```
***ABBREVIATION <abbreviation name> FORGOTTEN***

***REGION <region name> FORGOTTEN***

***REGION <region name 1>  FORGOTTEN***

***REGION <region name  2> FORGOTTEN***
```

One can only FORGET those things which are PRIVATE to himself.

Generally, only those REGIONS, ABBREVIATIONS, and FUNCTIONS which

have been SAVEd need be removed by using FORGET since items are automatically removed by NARIS at the end of a session if the user does not SAVE them.

One should be aware that REGIONS, ABBREVIATIONS, and FUNCTIONS take up space on the computer. Therefore, one should not SAVE everything--only those REGIONS, ABBREVIATIONS, and FUNCTIONS which he needs for future work. Also, all REGIONS, ABBREVIATIONS, and FUNCTIONS which are no longer useful to the user should be forgotten.

The use of FORGET may be thought of as being the same as SAVE with the opposite effect.

1. FORGET REGION SAVEXAMPLE1#

2. FORGET REGION SAVEXAMPLE2#

3. FORGET ABBREVIATION SAVEXAMPLE3., SAVEXAMPLETHREE.#

3.2.4 LIST

The NARIS language construct, LIST, is used to obtain a list of the following items: ABBREVIATIONS, REGIONS, FUNCTION, Data CLASSES, ELEMENTS, and VALUES.

The format of the LIST construct is

```
LIST PUBLIC#

    or

LIST SEMIPUBLIC#

    or

LIST PRIVATE#
```

or

LIST ABBREVIATIONS#

or

LIST REGIONS#

or

LIST FUNCTIONS#

or

LIST {PUBLIC or SEMIPUBLIC or PRIVATE} ABBREVIATIONS#

or

LIST {PUBLIC or SEMIPUBLIC or PRIVATE} REGIONS#

or

LIST {PUBLIC or SEMIPUBLIC or PRIVATE} FUNCTIONS#

or

LIST CLASSES#

or

LIST <class name> ELEMENTS#

or

LIST <data element name> VALUES#

If instructed to LIST ABBREVIATION, or REGIONS, or FUNCTIONS, NARIS will respond with a complete list of all PUBLIC, SEMIPUBLIC, and PRIVATE names. For example,

LIST REGIONS#

would result in NARIS responding with

THE PUBLIC REGIONS ARE:

{list of <region names>}

THE SEMIPUBLIC REGIONS ARE:

{list of <region names>}

THE PRIVATE REGIONS ARE:

{list of <region names>}

One may, instead of stating "LIST REGIONS#", request

LIST PUBLIC REGIONS#

in which case, only the PUBLIC <region names> would be listed:

THE PUBLIC REGIONS ARE:

{list of <region names>}

It must be noted that if there are no PUBLIC, SEMIPUBLIC, or PRIVATE - REGIONS, ABBREVIATIONS, or FUNCTIONS - that category would not be mentioned in the NARIS response. For example,

LIST ABBREVIATIONS#

when there are no SEMIPUBLIC ABBREVIATIONS would result in

THE PUBLIC ABBREVIATIONS ARE:

{list of <abbreviation names>}

THE PRIVATE ABBREVIATIONS ARE:

{list of <abbreviation names>}

Also, if one were to ask NARIS to LIST something that had no names,

it would answer with "***DONE***". For example, if there were no PUBLIC FUNCTIONS and one stated

```
LIST PUBLIC FUNCTIONS#
```

NARIS would answer with

```
***DONE***
```

The request

```
LIST PUBLIC#
```

would result in the NARIS response

```
THE PUBLIC REGIONS ARE:
     (list of <region names>)
THE PUBLIC ABBREVIATIONS ARE:
     (list of <abbreviation names>)
THE PUBLIC FUNCTIONS ARE:
     (list of <function names>)
```

LISTing SEMIPUBLIC or PRIVATE would result in the same NARIS response as shown above with the word "PUBLIC" replaced by the word which was used. Again, if there were, for example, no PUBLIC FUNCTIONS, performing a "LIST PUBLIC#" would result in only the PUBLIC REGIONS and ABBREVIATIONS being LISTed.

Some examples of the LIST construct used with REGIONS, ABBREVIATIONS, and FUNCTIONS are:

```
LIST ABBREVIATIONS#
```

```
LIST PRIVATE REGIONS#
LIST SEMIPUBLIC#
```

The LIST construct is also used to access the ELEMENTS, <data element names>, and VALUES in the system. For example,

```
LIST CLASSES#
```

would produce a NARIS response of

```
THE DATA CLASSES ARE
     (list of data <class names>)
```

From this list of data <class names>, one may decide to have NARIS list the <data element names> of one of the classes. For example,

```
LIST SOIL ELEMENTS#
```

would result in a NARIS response of the following <data element names>

```
SOIL SUMACRES
SOIL DATE
SOIL NUMBER
SOIL SLOPE
SOIL EROSION
SOIL ACRES
SOIL OVERLAP
```

and from this LIST of <data element names> one might request

```
LIST SOIL SLOPE VALUES#
```

which would result in a LIST of the data VALUES for the <data element

name> "SOIL SLOPE":

A

B

C

D

E

F

G

Some examples of the LIST construct being used to list CLASSES, ELEMENTS, and VALUES are:

LIST CLASSES#

LIST FORESTRY ELEMENTS#

LIST FORESTRY COVERTYPE VALUES#

LIST FUTUREIMPOUNDMENT ELEMENTS#

LIST GEOSURFICIAL TYPE VALUES#

3.2.5 WHAT IS (ARE)

Textual information pertaining to <region names>, <abbreviation names>, <function names>, and the data may be obtained by using the "WHAT IS" or "WHAT ARE" language construct.

The format of the "WHAT IS" construct is

WHAT IS REGION <region name>#

or

WHAT IS ABBREVIATION <abbreviation name>#

or

WHAT IS FUNCTION <function name>#

or

WHAT IS <class name>#

or

WHAT IS <data element name>#

or

WHAT IS <data element name> <value>#

or

WHAT ARE CLASSES#

or

WHAT ARE <class name> ELEMENTS#

or

WHAT ARE <data element name> VALUES#

Again, it is permitted to use a series of names when using "WHAT IS" with REGION, ABBREVIATION, and FUNCTION. For example,

WHAT IS REGION MARENGOTWP, FORESTRYINMARENGO, ABBREVIATION TOTRSCS.#

One may also request

WHAT IS SOIL, FORESTRY, WATERSHED REGIONALBASIN, SOIL SLOPE A#

NARIS would respond with textual descriptions for

SOIL,

FORESTRY,

WATERSHED REGIONALBASIN, and

SOIL SLOPE A.

It is not permitted to interrogate ("WHAT IS") REGIONS, ABBREVIATIONS, and FUNCTIONS with data. For example, this request is not allowed:

WHAT IS REGION SUBDIVISION, SOIL SLOPE G#

It should be noted that all <data element name> <values> are valid in a "WHAT IS" request. For example, attempting to obtain a description of an integer VALUE is not allowed - an error message would be given.

The request

WHAT ARE SOIL ACRES VALUES#

would result in

TEXT IS NOT STORED FOR THE VALUES OF SOIL ACRES

attempting to ask

WHAT IS FORESTRY ACRES 5#

would get the same error message.

The request

WHAT ARE CLASSES#

will produce textual descriptions for all data CLASSES in the NARIS System.

The request

WHAT ARE SOIL ELEMENTS#

will result in descriptions of each <data element name> being given. For example,

SOIL SLOPE IS:

AN INDICATOR OF THE TYPICAL SLOPE OF THE PLOT OF SOIL.

SOIL DATE IS:

THE YEAR OF THE RELEASE OF THE CORRELATED SOIL SURVEY BY SCS
FROM WHICH THE SOIL DATA IS TAKEN.

(the rest of the <data element names> and their descriptions
for Data Class, SOIL)

the request

WHAT IS REGION MARENGOTWP#

produces a response with the following information

(statement used to create the item,

name of person who created it,

date of its creation, and

number of times that it has been used)

Some examples of the "WHAT IS" construct are:

WHAT IS ABBREVIATION SLIGHTURBANLIMIT., REGION SUBDIVISION#

WHAT IS FUNCTION TIMBERCOST#

WHAT IS FUTUREIMPOUNDMENT#

WHAT IS GEOSURFICIAL TYPE#

WHAT IS GEOWASTE ACRES#

WHAT IS WATERSHED TRIBUTARY#

WHAT IS GEOSURFICIAL TYPE 5D#

WHAT IS SOIL NUMBER 353#

WHAT IS SOIL NUMBER BA#

WHAT ARE CLASSES#

WHAT ARE FORESTRY ELEMENTS#

WHAT ARE FORESTRY COVERTYPE VALUES#

WHAT ARE SOIL NUMBER VALUES#

WHAT ARE GEOSURFICIAL TYPE VALUES#

WHAT ARE LANDUSE1 CODE VALUES#

WHAT IS LANDUSE1 CODE 81.0#

The reader is encouraged to submit many of these example requests to NARIS in order to become familiar with the ways in which the data can be interrogated for textual descriptions.

3.3 Arithmetic Expressions

The use of arithmetic expressions allows the user to perform mathematical calculations on the data in the NARIS System.

There are two kind of arithmetic expressions:

1) Class Arithmetic Expressions; and

2) Tract Arithmetic Expressions

The choice of the arithmetic expression one uses depends on the language construct used.

3.3.1 Class Arithmetic Expression

A <class arithmetic expression> is used to perform calculations on Data Element Values.

3.3.1.1 Definition

There are five ways to define a <class arithmetic expression>:

1) ...<data element name>...#

where <data element name> must be the name of a numeric Data Element. (see 3.3.1.3)

2) ...<number>...#

with at least one <data element name> present.

3) ...<function name> (<class arithmetic expression>)...#

where <class arithmetic expression> is here defined as a <data element name> or a <number> or a (<class arithmetic expression>) or a <function name> (<class arithmetic expression>) or as noted in definition [illegible]).

4) ...(<class arithmetic expression>)...#

where <class arithmetic expression> is here de[illegible]

as a <data element name> or a <number> or a <function name> (<class arithmetic expression>) or a (<class arithmetic expression>) or as noted in definition 5).

5. ... <class arithmetic expression> { <arithmetic operator> or ...# }

where <class arithmetic expression> is here defined as being a <data element name> or a <number> or a <function name> (<class arithmetic expression>) or a (<class arithmetic expression>) or as in definition 5). Note that in 5), <class arithmetic expression> <class arithmetic expression> is not allowed, nor may a <class arithmetic expression> end with an <arithmetic operator>; e.g. ...SOIL ACRES+...# is not allowed.

The concept of "recursiveness" applies to the definitions of <class arithmetic expression> such that each definition is heavily dependent on the other definitions and its own definition.

The dots (...) refer to the context in which the <class arithmetic expression> is used in a language construct; thus, a <class arithmetic expression> does not constitute a complete NARIS request.

3.3.1.2 Numbers and Arithmetic Operators

The <arithmetic operators> used in <class arithmetic expressions> are:

Name:	<arithmetic operator>
addition :	+
subtraction :	-
multiplication :	*
division :	/
exponentiation :	**

<numbers> used in <class arithmetic expressions> must be either integers or decimal numbers. The following are examples of <numbers>:

0
10
.123
0.123
10.15
10.0
15.

3.3.1.3 Numeric Data Elements

All <data element names> used in <class arithmetic expressions> must have numerical values. Thus, one may request

...SOIL ACRES/10...#

but not

...SOIL NUMBER/10...#

as the values of SOIL NUMBER are non-numeric. The NARIS Data Guide (Appendix B) is constructed in such a manner that numeric Data Elements are distinctly noted from other Data Elements. All <data element names> used in a <class arithmetic expression> must be of the same Data Class. Thus,

...SOIL ACRES + (FORESTRY SUMACRES*10)...#

is not an allowed <class arithmetic expression>.

It should be noted that <data element names> must contain the <class name> of the Data Class. However, the <class name> may, if desired, be used only once in a request - when it is a part of the first or left-most

<data element name>. For example, the following two <class arithmetic expressions> are interpreted by NARIS as being the same:

...SOIL ACRES/SOIL SUMACRES*100...# and

...SOIL ACRES/SUMACRES*100...#

Thus, repetition of the <class name> in <data element names> is permitted but not necessary.

An example of incorrect <data element names> is

TABULATE ACRES/SUMACRES*100#

as there is no <class name> to complete the <data element names>. It should also be noted that the following two <class arithmetic expressions> are not valid because <data element name> is not present:

FOR TRACT TABULATE 10+5/10**2# and

FOR TRACT TABULATE SOIL 10+5/10**2#

3.3.1.4 Order of Numerical Calculations

Complex <class arithmetic expressions> may be interpreted by NARIS in a manner which was not meant by the user unless parentheses are used to specify the order in which calculations take place. In the absence of parentheses, the order or precedence of calculations is

first exponentiations,
then multiplications and divisions, and
finally additions and subtractions.

These rules cause the following two <class arithmetic expressions>

to be equivalent in meaning:

...SOIL ACRES/40+10*ACRES**2...#

...(SOIL ACRES/40)+(10*(ACRES**2))...#

Parentheses may be used to override the normal precedence of calculations. Thus, for the <class arithmetic expression>,

...SOIL ACRES*(10-SUMACRES)...#

the subtraction is performed before the multiplication.

When arithmetic operations have the same precedence, such as multiplication and division, the order of calculations is left to right as they appear in the expression. Thus,

...SOIL ACRES/SUMACRES*100...#

means (SOIL ACRES/SUMACRES)*100, not SOIL ACRES/(SUMACRES*100).

A final restriction on the use of the <arithmetic operators>, +, -, *, /, and **, is that none of these operators may appear side by side. Thus, SOIL ACRES*-10 is not a legal <class arithmetic expression>, but SOIL ACRES*(-10) is legal.

Figure III-18 on the following page, shows other correct and incorrect <class arithmetic expressions>.

Mathematical Notation	Correct Expression	Incorrect Expression
1) $A \cdot B$	A*B	AB no multiply operator
2) $A \cdot (-B)$	A*(-B)	A*-B two operators side by side
3) $-(A+B)$	-(A+B)	-A+B = (-A) + B
4) A^{B+2}	A**(B+2)	A**B + 2 = A^B + 2
5) $A+\frac{B}{10}$	A+(B/10)	(A+B)/10
6) $A(B+A)$	A*(B+A)	A(B+A) no multiply operator
7) $A+[B*A]$	A+(B*A)	A+[B*A]

Note: "A" and "B" are used as shorthand notations for two numeric Data Elements within the same data class.

Figure III-18. Examples of Correct and Incorrect Class Arithmetic Expressions.

3.3.1.5 Use in WHERE, TABULATE, and CALCULATE

To aid in describing the use of <class arithmetic expressions> with the WHERE <where clause> and the TABULATE and CALCULATE language constructs, an example $\frac{1}{4}$ $\frac{1}{4}$ section with the <region name> of TRACT will be used.

For purposes of explaining arithmetic expressions, TRACT will contain selected SOIL and FORESTRY Data Elements and values as noted in the following TABULATE request:

```
FOR TRACT TABULATE SOIL NUMBER, ACRES, SLOPE, SUMACRES, EROSION,
FORESTRY  COVERTYPE, ACRES, SUMACRES#
```

producing the following output

```
SOIL:
NUMBER     ACRES     SLOPE        SUMACRES        EROSION

W103           5       A             40              0
  27          25       A             40              1
W103          10       B             40              2

FORESTRY:
COVERTYPE             ACRES                SUMACRES

   MS                  10                     25
   CS                  15                     25
```

An example of a <class arithmetic expression> would be:

```
...(SOIL ACRES + 10)...#
```

In order to list the percentage of SOIL ACRES in TRACT which have a SOIL type of W103, one could submit a TABULATE request as follows:

```
FOR TRACT TABULATE SOIL ACRES, SUMACRES, ACRES/SUMACRES*100
WHERE NUMBER IS W103#
```

which would yield:

```
SOIL:
ACRES          SUMACRES          ACRES/SUMACRES*100

  5               40                  12.5
 10               40                  25.0
```

<class arithmetic expressions> may also be used in WHERE <where clauses> and CALCULATE requests. For example, the request:

```
FOR TRACT TABULATE SOIL NUMBER, (ACRES/SUMACRES)*100 WHERE
(ACRES/SUMACRES)*100 IS GREATER THAN 15#
```

would result in:

```
SOIL:
NUMBER              (ACRES/SUMACRES)*100

  27                      62.5
W103                      25.0
```

and the request:

```
FOR TRACT CALCULATE AVERAGE (SOIL ACRES/SUMACRES)*100 BY SOIL
NUMBER#
```

would result in

```
SOIL NUMBER         AVERAGE OF (SOIL ACRES/SUMACRES)*100

    W103                      18.75
      27                      62.5
```

3.3.2 Tract Arithmetic Expression

To produce MAPped output and OUTPUT to disk, it is necessary for NARIS to be able to assign a single value to a $\frac{1}{4}$ $\frac{1}{4}$ section. <tract arithmetic expression> is used to produce a single value for a $\frac{1}{4}$ $\frac{1}{4}$ section; whereas <class arithmetic expression> produces a single value for each occurrence in the $\frac{1}{4}$ $\frac{1}{4}$,

3.3.2.1 Definition

There are five ways to define <tract arithmetic expression>. As with <class arithmetic expression>, the definitions are recursive.

```
1)    { SUM or TOTAL   }
      {    or          }                                { WHERE <where clause> }
      { AVE or AVERAGE }                                {                      }
      {    or          }  <class arithmetic expression> {                      } ...#
...   { MIN or MINIMUM }                                {          or          }
      {    or          }                                {                      }
      { MAX or MAXIMUM }                                {       <empty>        }
      {    or          }
      {   COUNT        }
```

This definition of <tract arithmetic expression> is called a "variable". It comprises a <class arithmetic expression> preceded by the word SUM, TOTAL, AVE, AVERAGE, MIN, MINIMUM, MAX, MAXIMUM, or COUNT and may be followed by a WHERE <where clause>.

2) ...<number>...#

Contrary to a <class arithmetic expression>, <number> is a <tract arithmetic expression> - it does not have to have a <data element name> present.

3) ...<function name> (<tract arithmetic expression>)...#

Where <tract arithmetic expression> is here defined as being the definitions 1), 2), 3), 4), and 5).

4) ...(<tract arithmetic expression>...#

Where <tract arithmetic expression> is here defined to be 1), 2), 3), 4), and 5).

```
5)  ┌─────────────────────────────────────────┐
    ▼                                ┌ <arithmetic operator> ┐
 ...<tract arithmetic expression>    │          or           │
                                     └         ...#          ┘
```

Where <tract arithmetic expression> is here defined as being definitions 1), 2), 3), 4), and 5). The arrow from <arithmetic operator> to the beginning of <tract arithmetic expression> is intended to show that a "series" of "<tract arithmetic expression> <arithmetic operator> <tract arithmetic expression>" is allowed. However, " <tract arithmetic expression> <tract arithmetic expression>" and "<tract arithmetic expression> <arithmetic operator>" is not allowed.

3.3.2.2 Numbers and Arithmetic Operators

The <arithmetic operators> and <numbers> used in <tract arithmetic expressions> are the same as those used in <class arithmetic expressions>:

Name	<arithmetic operators>
addition	+
subtraction	-
multiplication	*
division	/
exponentiation	**

The following are valid <numbers>:

0
10
.123
0.123
10.15
10.0
15.

3.3.2.3 Variables

Variables, as defined in definition 1), can produce different values for a $\frac{1}{4}\frac{1}{4}$. For example, given a $\frac{1}{4}\frac{1}{4}$ which has the following SOIL data:

SOIL: NUMBER	ACRES	SLOPE	EROSION
W103	10	A	0
27	5	B	1
W103	5	A	2
W103	5	B	1
27	15	B	3

...TOTAL SOIL ACRES...#

equals 10+5+5+5+15=40.0

...AVE SOIL ACRES...#

equals (10+5+5+5+15)/5=8.0

...MIN SOIL ACRES...#

equals minimum of (10,5,5,5,15)=5.0

...MAX SOIL ACRES...#

equals maximum of (10,5,5,5,15)=15.0

...COUNT SOIL ACRES...#

equals 1+1+1+1+1=5.0

TOTAL (or SUM) sums up each of the SOIL ACRES values, AVE (or AVERAGE) divides the sum of all SOIL ACRES values by the number of values summed, MIN (or MINIMUM) finds the smallest SOIL ACRES value, MAX (or MAXIMUM) finds the largest SOIL ACRES value, and COUNT gives a count of the number of SOIL ACRES occurrences.

The occurrences which a "variable" will use tc calculate the final value for the $\frac{1}{4}\frac{1}{4}$ can be restricted by using a WHERE <where clause>.

For example,

```
...SUM SOIL ACRES WHERE SLOPE EQUALS B...#
```

equals $5+5+15=25.0$

Only those SOIL entries that have a SLOPE equal to B are SUMmed. The reader should study the section on WHERE clauses to determine how a WHERE clause is formed.

If a variable contains a WHERE clause such that when it is evaluated none of the occurrences satisfy the WHERE clause, a value of 0 is assumed for the $\frac{1}{4}$ $\frac{1}{4}$. For example, the variable

```
...SUM SOIL ACRES WHERE SLOPE EQUALS C...#
```

when evaluated on the example data, notes that there are no SOIL occurrences with SLOPE equal to C. Therefore, the value of the variable will be zero.

<class arithmetic expression> is used within the definition of variable. Thus the following are legal variables:

```
...SUM ((SOIL ACRES/40)*100)...#
```

equals $(\frac{10}{40}*100)+(\frac{5}{40}*100)+(\frac{5}{40}*100)+(\frac{15}{40}*100)=100.0$

```
...AVE (SOIL ACRES/EROSION)**2 WHERE SLOPE IS B...#
```

equals $((\frac{5}{1})^2+(\frac{5}{1})^2+(\frac{15}{3})^2)/3=25.0$

3.3.2.4 Order of Numerical Calculations

The rules which govern the order of calculations which take place within a <tract arithmetic expression> are the same as those for a <class arithmetic expression>. With the absence of parentheses, the order of calculations is:

first	exponentiations,
then	multiplications and divisions, and
finally	additions and subtractions.

Parentheses may be used to override these rules. Thus,

```
...10*((MAX SOIL ACRES)-(MIN SOIL ACRES))...#
```

cause the subtraction to be done before the multiplication.

When arithmetic operations have the same precedence, such as multiplication and division, the calculations are performed left to right as they appear in the expression. Thus,

```
...(TOTAL FORESTRY ACRES)/(TOTAL SOIL ACRES)*100...#
```

means

```
...((TOTAL FORESTRY ACRES)/(TOTAL SOIL ACRES))*100...#
```

<u>not</u>

```
...(TOTAL FORESTRY ACRES)/((TOTAL SOIL ACRES)*100)...#
```

3.3.2.5 Use in MAP and OUTPUT

MAP and OUTPUT are the NARIS language constructs in which <tract arithmetic expressions> are used. It should be noted, also, that MAP or OUTPUT may not be used without using a <tract arithmetic expression>.

The following examples of <tract arithmetic expressions> may be used

in either a MAP or an OUTPUT request (represented in the examples by "...").

Given data obtained from a TABULATE of a $\frac{1}{4}\frac{1}{4}$ section as

SOIL: NUMBER	ACRES	SLOPE
W103	5	A
27	25	A
W103	10	B

FORESTRY: COVERTYPE	ACRES	SUMACRES
MS	10	25
CS	15	25

Examples of <tract arithmetic expressions> are:

...TOTAL (FORESTRY ACRES)...#

equal 10+15=25.0

...(TOTAL (FORESTRY ACRES))+10...#

equals (10+15)+10=35.0

...(TOTAL ((FORESTRY ACRES/SUMACRES)*100))**2...#

equals $(\frac{10}{25}*100+\frac{15}{25}*100)^{2}=10000.0$

...(-(TOTAL (FORESTRY ACRES))/(TOTAL (SOIL ACRES)))*100...#

equals $\frac{-(10+15)}{(5+25+10)}*100=-62.5$

...(SUM (FORESTRY ACRES/25))*(AVE (FORESTRY ACRES) WHERE COVERTYPE IS MS...#

equals $(\frac{10}{25}+\frac{15}{25})*(\frac{10}{1})=10.0$

3.3.3 FUNCTIONS

A FUNCTION is used to assign a new value to the value produced by a calculation.

A FUNCTION can be thought of as an equation (on an X, Y axis) whose graph consists of a collection of line segments.

3.3.3.1 Definition

FUNCTIONS are created as shown in section 3.1.3 in the manual. For example, the function statement for defining TREECUTCOST would be:

THE FUNCTION TREECUTCOST IS (0,0) (50, 150) (100, 200)#

The following graph displays TREECUTCOST representing the cost of removing forestry from a $\frac{1}{4}\frac{1}{4}$ section of land:

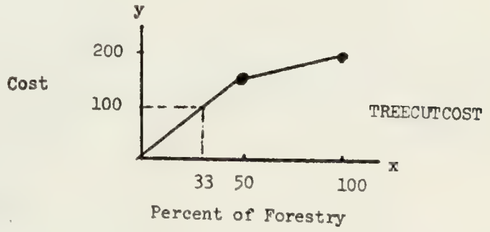


Figure III-'19. FUNCTION (y=f(x))

If a tract of land is 33% covered with forestry, the cost of clearing the land would be 100.

The x-coordinates are 0, 50, and 100 while the y-coordinates are 0, 150, and 200 respectively. Coordinate pairs must be listed so that the x-coordinate values are in increasing numeric order from left to right. The following definition of TREECUTCOST is not permitted and will cause NARIS to return an error message:

FUNCTION TREECUTCOST IS (0,0) (100,200) (50, 150)#

because the x-coordinate, 100, appears before the x-coordinate, 50. NARIS will process up to 15 coordinate pairs in a single FUNCTION statement; at least two coordinate pairs are required to define a line segment in a FUNCTION.

FUNCTIONS whose line segments are not all connected because they contain "jumps" can be defined. For example, the following FUNCTION attaches one of three possible values to calculation. Such FUNCTIONS are useful for producing weights whose values depend on exceeding a threshold.

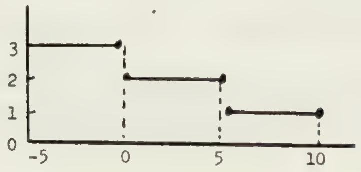


Figure III- 20. Example of a discontinous or "Jump" FUNCTION.

The request which would create the FUNCTION in the figure and give it

the <function name> "WEIGHTS" is

FUNCTION WEIGHTS IS (-5,3) (0-,3) (0,2) (5,2) (5+,1) (10,1)#

The "-" sign after the coordinate 0 in the coordinate pair (0-,3) states that "y" value 3 is not given to a calculated value of 0. The coordinate pair (0,2) states that the "y" value of 2 is to be attached to a calculated value of 0. Therefore, in the line segment (-5,3) (0-,3) the "y" value 3 is given to all the "x" values between -5 and 0, including -5, but not including 0. Similarly, in the line segment (5+,1) (10,1), the "y" value 1 is assigned to all calculated "x" values between 5 and 10, including 10, but not including 5. It is important to note that 5+ is a number that is slightly larger than 5 and 0- is a number slightly smaller than 0. Therefore, the coordinate (5+,1) cannot be written before (5,2) and (0,2) cannot be written before (0-,3).

3.3.3.2 FUNCTIONS in Class Arithmetic Expressions

For the purpose of creating example FUNCTIONS to be used in <class arithmetic expressions>, the following data is assumed to be a part of the REGION, TRACT:

FORESTRY: COVERTYPE	ACRES
MS	10
CS	15

Three functions representing the graphs:

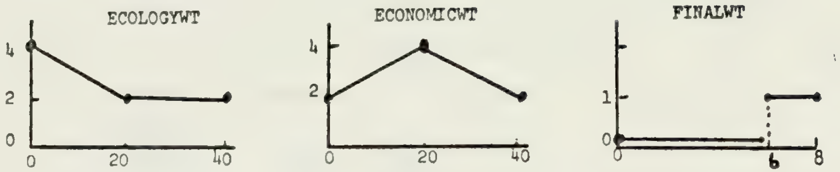


Figure III-21. FUNCTION Graphs of ECOLOGYWT, ECONOMICWT, and FINALWT

are created by these statements:

```
FUNCTION ECOLOGYWT IS (0,4) (20,2) (40,2)#

FUNCTION ECONOMICWT IS (0,2) (20,4) (40,2)#

FUNCTION FINALWT IS (0,0) (6-,0) (6,1) (8,1)#
```

More than one FUNCTION may be used in a request. For example,

```
FOR TRACT TABULATE FORESTRY ACRES, (ECOLOGYWT(FORESTRY ACRES)),
(ECONOMICWT(FORESTRY ACRES*2)), (FINALWT(ECOLOGYWT(FORESTRY ACRES)
+ECONOMICWT(FORESTRY ACRES*2)))#
```

will result in:

```
FORESTRY:
ACRES      (ECOLOGYWT(FORESTRY ACRES))        (ECONOMICWT(FORESTRY ACRES*2))

 10                3.0                                  4.0
 15                2.5                                  3.0
```

```
(FINALWT(etc.))...

     1.0
     0.0
```

It should be noted that when TABULATing a <class arithmetic expression> that begins with a FUNCTION, the entire expression must be enclosed in parentheses. If parentheses are not used, an error message will be produced which says"<function name> IS NOT A DATA ELEMENT IN <class name> CLASS".

One should also note that if an expression is calculated and the resulting value to be operated upon by a FUNCTION is not within the range of the FUNCTION, an error message will be sent to the user. For example,

```
FUNCTION WEIGHT IS (0,0) (5,5)#

FOR TRACT TABULATE (WEIGHT (FORESTRY ACRES))#
```

cannot be executed because the FORESTRY ACRES <data element values> are 10 and 15 but the FUNCTION is only defined for the range 0 to 5. An error message would be generated which says "ARGUMENT ABOVE FUNCTION RANGE."

FUNCTIONS may also be used without being defined by a <function name>. Instead of:

```
FUNCTION ECOLOGYWT IS (0,4) (20,2) (40,2)#

FOR TRACT TABULATE (ECOLOGYWT (FORESTRY ACRES))#
```

one may request:

```
FOR TRACT TABULATE (FUNCTION [(0,4) (20,2) (40,2)] (FORESTRY ACRES))#
```

In the first case the definition of the <function name> is remembered and can be used in other arithmetic expressions by including its name, ECOLOGYWT. In the second case, the FUNCTION is not remembered.

3.3.3.3 FUNCTIONS in Tract Arithmetic Expressions

FUNCTIONS in <tract arithmetic expressions> are used exactly the same as described for <class arithmetic expressions>.

Consider, the following data in a REGION:

SOIL: NUMBER	ACRES	SLOPE
W103	5	A
27	10	A
W103	25	B

FORESTRY: COVERTYPE	ACRES
MS	10
CS	15

and the following FUNCTION definitions:

```
FUNCTION ECOLWT IS (0,0) (20-,0) (20,100) (40,100)#

FUNCTION ECONWT IS (0,0) (20,100) (40,150)#

FUNCTION FINALWT IS (0,0) (1000,500)#
```

From the data and definitions above, examples of using FUNCTIONS in <tract arithmetic expressions> are:

```
...ECOLWT(TOTAL FORESTRY ACRES)...#

    equals 100.0

...FUNCTION [(0,0) (1000,-1000)]((TOTAL FORESTRY ACRES)**2)...#
```

```
        equals -625.0

...FINALWT(ECONWT(TOTAL FORESTRY ACRES))...#

        equals 56.25

...0.25*ECONWT(TOTAL SOIL ACRES WHERE SLOPE IS A)+0.75*ECOLWT
(TOTAL FORESTRY ACRES)...#

        equals 93.75

...FINALWT(TOTAL(FUNCTION[(0,0) (10,0) (40,30)](FORESTRY ACRES)))...#

        equals 2.5
```

3.3.4 Possible Execution Errors when using Arithmetic Expressions

3.3.4.1 Zero Denominator

A division by zero is not mathematically defined. Consider the request

```
FOR TRACT TABULATE SOIL ACRES, EROSION#
```

resulting in:

SOIL! ACRES	EROSION
5	0
25	1
10	2

The first occurrence of SOIL in TRACT has an acreage value of 5 and an erosion value of 0. Thus, the request:

```
FOR TRACT TABULATE SOIL ACRES/EROSION#
```

will recognize that 5 divided by 0 cannot be completed. NARIS will send the user an error message:

```
ERRORS OCCURRED DURING RETRIEVAL.  THE TRACTS CAUSING THEM WERE
PUT IN AN ERROR REGION.
SOME TRACTS HAD DIVIDE BY ZERO ERRORS
```

3.3.4.2 Exponent Underflow and Overflow

Values produced in the evaluation of exponentiations should not exceed 10^{60} or 10^{-57}. If the value exceeds one of these limits, NARIS will return an error message.

From the SOIL data presented in 3.3.4.1, one could construct a request which would produce an exponent overflow message by

```
FOR TRACT TABULATE SOIL ACRES**70#
```

NARIS would respond with

```
ERRORS OCCURRED DURING RETRIEVAL.  THE TRACTS CAUSING THEM WERE
PUT IN AN ERROR REGION.
SOME TRACTS HAD EXPONENT OVERFLOW ERRORS
```

and the request

```
FOR TRACT TABULATE SOIL ACRES**(-60)#
```

would produce the same response as above, with the exception that the word "UNDERFLOW" would replace the word "OVERFLOW."

3.4 WHERE <where clause>

The WHERE <where clause> may be used to create a REGION (see 3.1.1.3) or to search each data occurrence in the REGION or each ¼ ¼ of a REGION for specific values in order to TABULATE, CALCULATE, MAP, or OUTPUT.

The WHERE <where clause> is the tool used to "interrogate" a geographic area (REGION) regarding its data content.

3.4.1 Definition of WHERE <where clause>

WHERE is a reserved word which must precede a <where clause>:

```
...WHERE <where clause>...#
```

<where clause> may be defined six (6) ways:

1) <data element name> <relational operator> <data element value>

   Note that <data element name> must represent numeric <data element values> except when the <relational operator> EQ or NE is used.

2) <data element name> IS { NOT or <empty> } ONE OF (▼<data element value> ,)

3) <class arithmetic expression> <relational operator> <class arithmetic expression>

4) <class arithmetic expression> IS { NOT or <empty> } ONE OF (▼ <integer> ,)

5) (<where clause>)

   <where clause> is defined as in 1), 2), 3), 4), 5) or 6).

```
     ┌──────────────────┐
     ▼                  │
6)  <where clause> { {AND} }
                     {OR }
                      or
                   <empty>
```

Here, <where clause> is defined as in 1), 2), 3), 4), 5) or 6). Note that the constructs "<where clause> <where clause>" and "<where clause> AND" (or OR) are not allowed.

3.4.2 Relational Operators

The following chart comprises the <relational operators> which may be used in a <where clause>:

<relational operators>

GT	or	GTR	or	GREATER THAN
GE	or	GEQ	or	GREATER THAN OR EQUAL TO
LT	or	LSS	or	LESS THAN
LE	or	LEQ	or	LESS THAN OR EQUAL TO
EQ	or	EQL	or	EQUALS or EQUAL TO
NE	or	NEQ	or	DOES NOT EQUAL

The words, NOT or IS, may precede any of the above.

A great deal of latitude is permitted in the phrasing of a <relational operator>. For example, all of the following are different ways of expressing the <relational operator> (≦):

IS LESS THAN OR EQUALS
LE
LEQ
NOT GT
LT OR EQ
LT OR EQUAL
NOT GREATER
NOT GE OR IS EQUAL
IS LESS OR EQUAL
LESS OR IS EQUAL TO
IS NOT GREATER THAN

3.4.3 WHERE <where clause> with <data element name>

Using definition 1) one can obtain a listing of all $\frac{1}{4}$ $\frac{1}{4}$'s in MARENGOTWP which have a GEOSANDGRAVEL TYPE of BR (bedrock). According to definition 1), one can construct the <where clause> part of the request as

...GEOSANDGRAVEL TYPE EQ BR...#

and, adding the reserved word, WHERE,

...WHERE GEOSANDGRAVEL TYPE EQ BR...#

The entire TABULATE request would be

FOR MARENGOTWP TABULATE TOTRSCS. WHERE GEOSANDGRAVEL TYPE EQ BR#

Definition 2) is formed such that one may search a geographic area for one or more <data element values>. For example, if one wanted to know the number of acres and the percent of those acres in a REGION (MARENGOTWP) for the plots of SOIL which are denoted by NUMBERs 103, W103, S103, one could request

FOR MARENGOTWP CALCULATE TOTAL AND PERCENT OF SOIL ACRES WHERE
NUMBER IS ONE OF (103, W103, S103) BY NUMBER#

The <where clause> in this request is "NUMBER IS ONE OF (103, W103, S103)." It must be noted that NUMBER is a <data element name> because the <class name> has been presented earlier in the request and the <class name> applies to all Data Elements which follow it. The above request actually

contains three <data element names>:

- SOIL ACRES,
- SOIL NUMBER, and
- SOIL NUMBER.

The examples of <where clause> which follow apply specifically to definitions 1) and 2):

```
1.  FOR TRACTEXAMPLE1 CALCULATE TOTAL GEOSANDGRAVEL ACRES WHERE
    TYPE EQL BR#

2.  FOR TRACTEXAMPLE7 TABULATE LANDUSE1 WHERE CODE IS ONE OF (33.0,
    64.0, 71.0, 72.0)#

3.  FOR WHEREEXAMPLE3 MAP TOTAL FUTUREIMPOUNDMENT ACRES*2 WHERE
    SITENUMBER GTR 0#

4.  FOR WOODS OUTPUT ON FILE "NAMEUSER" MAXIMUM SOIL ACRES WHERE
    EROSION LT 2, TOTAL STREAMS FLOODACRES WHERE FLOODACRES GTR 0#
```

The WHERE <where clause> in the four previous examples are:

```
1.  "WHERE TYPE EQL BR" (GEOSANDGRAVEL)
2.  "WHERE CODE IS ONE OF (33.0, 64.0, 71.0, 72.0)" (LANDUSE1)
3.  "WHERE STREAMS FLOODACRES NOT GTR 0"
4.  "WHERE EROSION LT 2" (SOIL) and
    "WHERE FLOODACRES GTR 0" (STREAMS)
```

3.4.4 WHERE <where clause> with <class arithmetic expression>

Definitions 3) and 4) of the <where clause> widen the scope of the WHERE "search" to values which are the result of calculations. Thus, instead of searching a geographic area for conditions which apply to <data element values>, one may look for areas which satisfy calculations involving <class arithmetic expressions>. For example, from definition 3), one can request

```
FOR MARENGOTWP TABULATE LANDUSE1 WHERE SOIL ACRES GT (2/3)*
SOIL SUMACRES#
```

and from definition 4) one may construct a <where clause>

```
FOR WOODS TABULATE FORESTRY WHERE FORESTRY ACRES*2 IS ONE OF
(10,20,30)#
```

It should be noted that the values of <class arithmetic expressions> are rounded to integers before "ONE OF" comparisons.

Note specifically that in definition 3) both <class arithmetic expressions> must refer to the same Data Class. Thus, it is not allowed to request

```
...WHERE LANDUSE1 ACRES EQ SOIL ACRES...#
```

3.4.5 Recursive Definitions of WHERE <where clause>

Definition 5) means that one may enclose a <where clause> in parentheses or that one may enclose a <where clause> which is enclosed by parentheses with parentheses, etc. Recursion! For example,

```
FOR MARENGOTWP CALCULATE AVERAGE SOIL ACRES WHERE (SOIL ACRES
GT (2/3)* SOIL SUMACRES) BY NUMBER#
```

Definition 6) allows one to AND or OR <where clauses>. For example, one can request

```
FOR WOODS TABULATE SOIL WHERE ACRES GEQ 5 AND EROSION LT 2#
```

The two <where clauses> in this request are "ACRES GEQ 5" and "EROSION LT 2". The two clauses become one <where clause> by joining them with the word, AND.

The AND means that each "internal" <where clause> of the <where clause> must be satisfied before the specific occurrences of SOIL data within the REGION is TABULATEd. For example, given the following SOIL data on a $\frac{1}{4}$ $\frac{1}{4}$ in REGION WOODS,

SOIL: NUMBER	ACRES	EROSION
103	2	2
152	11	0
330	5	0
224	7	2
134	4	0
134	11	0

the <where clause> in the previous example would be satisfied (because of the 2nd, 3rd, and 6th occurrence) and the TABULATE of SOIL data for the 2nd, 3rd, and 6th occurrences would be performed.

If the word, OR, had been used rather than AND in the request, the TABULATE would have also been performed. In the case of OR, if either of the "internal" <where clauses>is evaluated as being satisfied for an occurrence, the WHERE <where clause> is considered to be satisfied and the action requested by the language construct will be performed. Thus, if the word, AND, were replaced by OR in the previous example, the 2nd, 3rd, 4th, 5th, and 6th occurrences would be TABULATEd.

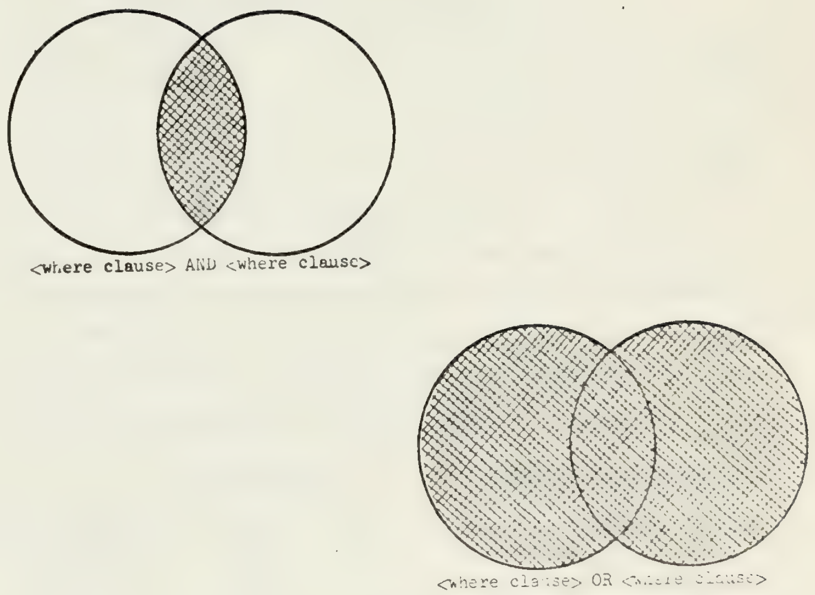


Figure III- 22. AND and OR in a <where clause>.

One should also note from definition 6) that the following WHERE <where clause> is valid

```
...WHERE SOIL NUMBER IS ONE OF (103,224) AND ACRES GT 5 OR
EROSION NEQ 2...#
```

The obvious question looking at the <where clause> above is "What does it mean?"

Without using parentheses to specify the order of the evaluation, the AND is evaluated first followed by the OR in a <where clause>. Therefore, the request above would be treated as if it were

```
...WHERE(SOIL NUMBER IS ONE OF (103, 224) AND ACRES GT 5) OR
EROSION NEQ 2...#
```

The AND and OR precedence may be superseded by the u:e of parentheses. For example,

```
...WHERE SOIL NUMBER IS ONE OF (103, 224) AND (ACRES GT 5 OR
EROSION NEQ 2)...#
```

A WHERE <where clause> which consists of <data element names> of a single class and which contains AND(s)and OR(s) is evaluated by Data Class occurrence. Thus, constructions like

```
...WHERE LANDUSE2 CODE IS 1 AND LANDUSE2 CODE IS 2...#
```

will never be satisfied, as each CODE data value denotes a different Data Class (LANDUSE2) occurrence.

When a WHERE <where clause> which contains <data element names> of more than one class and which contains AND(s) and OR(s) is evaluated, a <where clause> which is satisfied by a data occurrence causes the $\frac{1}{4}$ $\frac{1}{4}$ section to be included in the <new region> when creating a REGION or TABULATEd when TABULATing. Note that any internal <where clause> which contains AND(s) and OR(s) and <data element names> of a single class are dealt with by Data Class occurrence (see paragraph above).

3.4.6 Special Characteristics of a WHERE <where clause>

Creating a REGION by using a WHERE <where clause> produces a group of $\frac{1}{4}$ $\frac{1}{4}$ sections which have similar characteristics. The $\frac{1}{4}$ $\frac{1}{4}$ is tested to "satisfy" the <where clause>. If any occurrence satisfies the <where clause>, all data occurrences of the tract become a part of the REGION (see 3.1.1.3, REGION Request with WHERE clause).

A <where clause> may contain <data element names> of different classes only when used to form a REGION or when used in a TABULATE. In all other cases, the <where clause> must contain <data element names> of the same Data Class. The chart in Figure III-31 illustrates when more than one Data Class may be referenced in a <where clause>.

	One Data Class	More Than One Data Class
REGION creation	yes	yes
TABULATE	yes	yes
CALCULATE	yes	no
OUTPUT	yes	no
MAP	yes	no

Figure III-23. Number of Data Classes which may be used in a <where clause>.

3.4.7 Examples of WHERE <where clause>

The <where clauses> in the following example requests are underlined:

1. FOR MYREGION1 WHERE (SOIL ACRES GT 10 OR NUMBER IS ONE OF (102, 134)) AND FUTUREIMPOUNDMENT ACRES NOT GTR 0 TABULATE FORESTRY, GEOSURFICIAL, SOIL#

2. REGION MYREGION2 IS MYREGION1 WHERE STREAMS FLOODACRES NOT GT 0

AND IMPOUNDMENT ACRES EQ 0 AND FUTUREIMPOUNDMENT ACRES NOT GREATER THAN 0#

3. FOR MYREGION2 CALCULATE TOTAL AND PERCENT OF SOIL ACRES BY NUMBER WHERE NUMBER IS NOT ONE OF (W103, S103) AND EROSION EQL 0#

4. FOR MYREGION2 OUTPUT ON FILE "NAMEOFMYFILE" SUM (ECOWEIGHT (FORESTRY ACRES)*2) WHERE FORESTRY ADEQUATEMANAGEMENT EQ Y AND STANDIB EQ 1#

5. FOR MYREGION2 MAP ((SUM 2*LANDUSE1 ACRES WHERE CODE IS 11.1 AND COUNT GEQ 5)+50*(AVE(LANDUSE1 ACRES/40*100) WHERE CODE IS ONE OF (33.0, 72.0, 74.0)) - 10*(SUM STREAMS FLOODACRES WHERE FLOODACRES GTR 0))#

Note that example 5 contains three distinct <where clauses> - each being a component of the three internal <tract arithmetic expressions>.

3.5 Language Constructs which access the Data Base

The NARIS language constructs, TABULATE, CALCULATE, MAP, and OUTPUT allow one to access geographically stored data from REGIONS within NARIS.

3.5.1 TABULATE

A TABULATE request may be thought of as performing the following actions for each $\frac{1}{4}$ $\frac{1}{4}$ section:

1) it opens a "window" to look at the data in each data occurrence within the specified Data Class(es);

2) if the data attribute(s) specified in the TABULATE request is present, it is extracted;

3) if a calculation is to be performed, it is performed on the data values;

4) the information which has been requested is "written" to the user.

TABULATE produces a listing of <class names>, <data element names>, and <data element values> by $\frac{1}{4}$ $\frac{1}{4}$ section for the REGION being TABULATEd.

3.5.1.1 Definition

A TABULATE request is defined as

```
                  ┌──────────────────────────────────────────────────────,─┐
                  ▼                                                        │
             ┌ <class name>                   ┐   ┌ WHERE <where clause> ┐
             │      or                        │   │                      │
             │ <data element name>            │   │                      │
...TABULATE  │      or                        │   │         or           │  ...#
             │ <class arithmetic expression>  │   │                      │
             │      or                        │   │       <empty>        │
             └ <empty>                        ┘   └                      ┘
```

A TABULATE request, thus, may contain a <class name> or a <data element name> or a <class arithmetic expression>. TABULATE may also contain a WHERE <where clause>.

The arrow with the comma attached to it indicates that the items encompassed by the arrow may be used more than once, if separated by a comma. For example, one could

```
...TABULATE <class name 1> WHERE <where clause 1>, <data element
name 1>, <class name 2>, <class arithmetic expression> WHERE
<where clause 2>...#
```

Regarding the use of commas and WHERE <where clauses> in a TABULATE request, one should be cautioned that the comma may be considered a separator except - all items (even though separated by a comma) before a WHERE <where clause> will not be TABULATEd unless the <where clause> is satisfied.

For example, in the request above, the following action would be taken by NARIS:

1) If <where clause 1> was satisfied, <class name 1> would be TABULATEd;

2) Only if <where clause 2> was satisfied, would <data element name 1>, <class name 2>, and <class arithmetic expression> be TABULATEd.

When NARIS evaluates a <where clause> as false--that is, not true for a $\frac{1}{4}$ $\frac{1}{4}$ in a REGION - all items preceding the <where clause> and following an earlier <where clause> are not TABULATEd for that $\frac{1}{4}$ $\frac{1}{4}$ section.

3.5.1.2 The "empty TABULATE"

From the definition of the TABULATE request, one may

```
...TABULATE <empty> <empty>...#
```

and one may

```
...TABULATE <empty> WHERE <where clause>...# or

...TABULATE <empty> WHERE <where clause 1>, <empty> WHERE <where
clause 2>, {etc.}...#
```

The "empty TABULATE" produces a list of the $\frac{1}{4}$ $\frac{1}{4}$ sections which comprise the REGION. For example,

```
FOR MARENGOTWP TABULATE#
```

would produce a NARIS response of

```
NEQ NEQ SEC 1 T44N R5E
        :
(<tract specifications> for each 1/4 1/4 section thru)
        :
SWQ SWQ SEC 36 T44N R5E
```

Thus, one may obtain a list of all $\frac{1}{4}$ $\frac{1}{4}$'s which comprise a REGION by

```
FOR <existing region> TABULATE#
```

To obtain a list of $\frac{1}{4}$ $\frac{1}{4}$'s WHERE specific data attributes are present one may "TABULATE WHERE <where clause>". For example, if one wanted to know the <tract specifications> for all $\frac{1}{4}$ $\frac{1}{4}$'s in MARENGOTWP which contained FORESTRY COVERTYPE HN (mixed hardwoods), the following request could be used:

```
FOR MARENGOTWP TABULATE WHERE FORESTRY COVERTYPE EQ HN#
```

NARIS would respond with a list of $\frac{1}{4}$ $\frac{1}{4}$ sections which contained the <data element value>, HN, for FORESTRY COVERTYPE.

It should be noted that when a WHERE <where clause> is satisfied in a TABULATE, the <tract specification> will be followed by the <where clause> which caused the $\frac{1}{4}$ $\frac{1}{4}$ to be listed. For example, the response from NARIS would be of the format

```
NEQ NEQ SEC 1 T44N R5E
FORESTRY COVERTYPE EQ HN  SATISFIED
```

Each WHERE <where clause> that is satisfied on a $\frac{1}{4}$ $\frac{1}{4}$ will be listed after the $\frac{1}{4}$ $\frac{1}{4}$'s <tract specification>. The WHERE <where clauses> which were not satisfied on the $\frac{1}{4}$ $\frac{1}{4}$ will not be listed.

Some examples of the "empty TABULATE" request are

1) FOR SUBDIVISION TABULATE#

2) FOR MYREGIONNAME TABULATE#

3) FOR SUBDIVISION TABULATE WHERE FORESTRY ACRES NEQ 0 AND STANDNB EQL 1#

4) FOR SUBDIVISION TABULATE WHERE GEOSURFICIAL ACRES EQ 30, WHERE FUTUREIMPOUNDMENT ACRES NEQ O#

3.5.1.3 TABULATE <class name> and WHERE <where clause>

3.5.1.3.1 TABULATE <class name>

From the definition (3.5.1.1), one can

...TABULATE <class name>...#

or

...TABULATE <class name>, <class name>, {etc.}...#

<class name>, of course, is the name of any Data Class in NARIS.

The following are examples of TABULATE <class names>:

1) FOR SUBDIVISION TABULATE GEOSURFICIAL#

2) FOR SUBDIVISION TABULATE LANDUSE1, LANDUSE2, FORESTRY#

3) FOR SUBDIVISION TABULATE TOTRSCS.#

TABULATing <class names> results in all data attributes of the Data Class being listed.

3.5.1.3.2 TABULATE <class name> with WHERE <where clause>

Again, from the TABULATE definition in 3.5.1.1, one notes that a TABULATE request may occur as

...TABULATE <class name> WHERE <where clause>...#

or a series of <class names> and WHERE <where clauses> as

...TABULATE <class name>, <class name> WHERE <where clause>, <class name> WHERE <where clause>, <class name>...#

Some examples of TABULATE <class name> with WHERE <where clause> are:

1) FOR MARENGOTWP TABULATE GEOSURFICIAL WHERE TYPE EQL 8#

2) FOR MARENGOTWP TABULATE FORESTRY WHERE ACRES GTR 5, SOIL, WELL WHERE DEPTH GTR O#

3) FOR MARENGOTWP TABULATE SOIL WHERE NUMBER IS W AND ACRES GT 10, WHERE LANDUSE1 CODE IS 95.0#

3.5.1.4 TABULATE <data element name> and WHERE <where clause>

3.5.1.4.1 TABULATE <data element name>

From the TABULATE definition (3.5.1.1), a TABULATE request can be

formed as:

```
...TABULATE <data element name>...#
            or
...TABULATE <data element name>, <data element name>, (etc.)...#
```

The use of a TABULATE <data element name> will result in only data attributes of the <data element name> being listed for each $\frac{1}{4}$ $\frac{1}{4}$ in the REGION.

Some examples of TABULATE <data element names> are:

1) FOR MYREGION TABULATE SOIL NUMBER, ACRES#

2) FOR MYREGION TABULATE GEOWASTE TYPE#

3) FOR MYREGION TABULATE FORESTRY COVERTYPE, SOIL NUMBER, EROSION#

3.5.1.4.2 TABULATE <data element name> with WHERE <where clause>

One should note from the definition of TABULATE (3.1.1.1) that a WHERE <where clause> may be used with "TABULATE <data element name>," such that one can request:

```
...TABULATE <data element name> WHERE <where clause>...#
                  or
...TABULATE <data element name> WHERE <where clause>, <data element
name> WHERE <where clause> (etc.)...#
```

Thus, one may obtain a list of data attributes only WHERE a condition exists.

If the <data element name> is of the same Data Class as the <where clause>, the <data element name> will only be TABULATEd for those occurrences

which satisfy the <where clause>.

The following are examples of TABULATing <data element names> with WHERE <where clauses>:

1) FOR MYNEWREGION TABULATE SOIL ACRES WHERE NUMBER IS ONE OF (W103, S103)#

2) FOR MYNEWREGION TABULATE GEOWASTE TYPE, GEOWASTE ACRES, GEOSURFICIAL TYPE WHERE ACRES GTR 10#

3) FOR MYNEWREGION TABULATE SOIL NUMBER WHERE EROSION LEQ 2, STREAMS FLOODACRES WHERE FLOODACRES GTR 0#

3.5.1.5 TABULATE <class arithmetic expression> and WHERE <where clause>

3.5.1.5.1 TABULATE <class arithmetic expression>

The following TABULATE request is compatible with the definition of TABULATE (3.1.1.1):

```
...TABULATE <class arithmetic expression>...#
               or
...TABULATE <class arithmetic expression>, <class arithmetic
expression>, (etc.)...#
```

The output received from TABULATing a <class arithmetic expression> will provide the result of the calculation on the data attributes which appear in the $\frac{1}{4}$ $\frac{1}{4}$ section for the specified <data element name> of the <class arithmetic expression>.

Some examples of TABULATing <class arithmetic expressions> are:

1) FOR NEWREGIONNAME TABULATE SOIL ACRES/40*100#

2) FOR NEWREGIONNAME TABULATE LANDUSE1 CODE, LANDUSE1 ACRES/40*100#

3) FOR NEWREGIONNAME TABULATE (SOIL ACRES/SOIL SUMACRES)*100#

3.5.1.5.2 TABULATE <class arithmetic expression> with WHERE <where clause>

WHERE <where clauses> may be used in a TABULATE request which contains a <class arithmetic expression>, e.g.,

```
...TABULATE <class arithmetic expression> WHERE <where clause>...#

                              or

...TABULATE <class arithmetic expression> WHERE <where clause>,
<class arithmetic expression> WHERE <where clause>, (etc.)...#
```

Thus, one may request that a calculation be performed on data attributes WHERE a data condition is satisfied.

If the <class arithmetic expression> contains <data element names> of the same Data Class as the <where clause>, the TABULATE of the <class arithmetic expression> will be performed on only those occurrences which satisfy the <where clause>.

Some examples of TABULATing <class arithmetic expressions> with WHERE <where clauses> are:

1) FOR EXAMPLEREGION TABULATE SOIL ACRES/2 WHERE NUMBER IS ONE OF (152, 103, 348)#

2) FOR EXAMPLEREGION TABULATE SOIL NUMBER, (SOIL ACRES/SUMACRES)*100 WHERE EROSION GTR 1#

3) FOR EXAMPLEREGION TABULATE WHERE SOIL NUMBER IS NOT ONE OF (152, 348), LANDUSE1 ACRES/40*100 WHERE LANDUSE1 CODE IS NOT 11.1#

3.5.1.6 TABULATing <class names>, <data element names>, <class arithmetic expressions>, and WHERE <where clauses>

One may use any combination of <class names>, <data element names>, <class arithmetic expressions>, and WHERE <where clauses> in a TABULATE request. (See definition of TABULATE - 3.5.1.1)

From the following examples, the reader may want to use a REGION of his own and perform some of the TABULATEs.

1) FOR EXAMPLEREGION1 TABULATE SOIL, FORESTRY COVERTYPE, GEOWASTE TYPE WHERE ACRES GTR 10#

2) FOR EXAMPLEREGION1 TABULATE STREAMS WHERE FLOODACRES GTR 0, SOIL NUMBER WHERE ACRES GTR 5, (GEOSANDGRAVEL ACRES/SUMACRES) *100 WHERE TYPE EQL BR, LANDUSE1#

3) FOR EXAMPLEREGION1 TABULATE WHERE PLANTATION ACRES GTR 0#

4) FOR EXAMPLEREGION1 TABULATE GEOSURFICIAL TYPE WHERE ACRES GTR 10 AND TYPE IS NOT WA#

5) FOR EXAMPLEREGION1 TABULATE LANDUSE1, WHERE LANDUSE2 CODE IS ONE OF (1,2,3) AND ACRES GT 5#

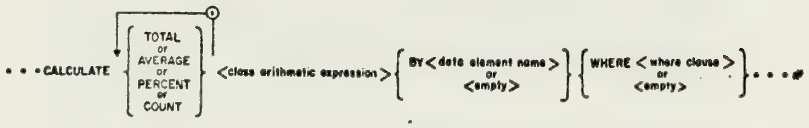
6) FOR EXAMPLEREGION1 TABULATE STREAMS, WELL, WHERE LANDUSE1 CODE EQL 95.0#

3.5.2 CALCULATE

CALCULATE is the language construct used to perform calculations on data attributes and list the answers BY another data attribute of the same Data Class. CALCULATE is used to calculate summary characteristics of a geographic area.

3.5.2.1 Definition

The definition of a CALCULATE request is shown in the following figure:



. . . CALCULATE
TOTAL or AVERAGE or PERCENT or COUNT
<class arithmetic expression>
BY <data element name> or <empty>
WHERE <where clause> or <empty>
. . . #

Figure III-24. CALCULATE request

The arrow is used to denote that all combinations of the words TOTAL, AVERAGE, PERCENT, and COUNT may be used - with the following restrictions:

1) Each word must be separated by a comma;

2) None of the four words may be used more than once in a CALCULATE request.

Thus, one may request

...CALCULATE TOTAL, PERCENT <class arithmetic expression>...#

but not

...CALCULATE TOTAL, AVERAGE, TOTAL <class arithmetic expression>...#

Also these words have other forms which may be used. For example, one may use

TOTAL or SUM
AVERAGE or AVE
PERCENTAGE or PERCENT

Therefore,

...CALCULATE SUM <class arithmetic expression>...#

and

...CALCULATE AVE, PERCENT <class arithmetic expression>...#

are valid CALCULATE requests.

3.5.2.2 CALCULATE without BY and WHERE

From the definition of CALCULATE (3.5.2.1), one notes the use of BY <data element name> and WHERE <where clause> are optional. This section presents CALCULATE requests which do not contain BY or WHERE. In the following example, REGION FORESTRYREGION includes four ¼ ¼'s (A, B, C, and D) which contain <data element values> as reflected below:

```
¼ ¼ A                              ¼ ¼ B
FORESTRY:                          FORESTRY:
    STANDNB 1                          STANDNB 1
    ACRES   10                         ACRES   5

¼ ¼ C                              ¼ ¼ D
FORESTRY:                          FORESTRY:
    STANDNB 1                          STANDNB 1
    ACRES   7                          ACRES   15
    STANDNB 2                          STANDNB 2
    ACRES   4                          ACRES   13
```

According to the CALCULATE definition of

```
...CALCULATE TOTAL <class arithmetic expression>...#
```

one can request

```
FOR FORESTRYREGION CALCULATE TOTAL FORESTRY ACRES#
```

FORESTRY ACRES is, of course, the <class arithmetic expression>.

- TOTAL is used to sum all FORESTRY ACRES <data element values> in FORESTRYREGION and produce one number as the result. The result of the "TOTAL" request, above is 54.
- AVERAGE is used to provide the number resulting when the number which would be provided by using TOTAL is divided by the number which would result from using COUNT. AVERAGE, then, is "TOTAL divided by COUNT". For the example REGION, AVERAGE FORESTRY ACRES would produce a value of 9.
- COUNT produces the number of occurrences of the <class arithmetic expression> in the REGION. A COUNT OF FORESTRY occurrences in FORESTRYREGION will produce a value of 6.
- PERCENT provides the percentage of the <class arithmetic expression> in the REGION. PERCENT will be 100 when neither a BY nor a WHERE is used in the CALCULATE request. For example, in FORESTRYREGION the PERCENT of FORESTRY ACRES IS ((54/54)*100)=100%.

For a request like

```
FOR MYREGION CALCULATE TOTAL FORESTRY ACRES#
```

NARIS will sum all FORESTRY ACRES in MYREGION - resulting in

```
TOTAL      OF FORESTRY ACRES
(<number> of acres of FORESTRY in MYREGION)
```

Using AVERAGE with the same <class arithmetic expression>, one can request

```
FOR MYREGION CALCULATE AVERAGE FORESTRY ACRES#
```

. resulting in

```
AVERAGE    OF FORESTRY ACRES
(average <number> of FORESTRY ACRES per occurrence of  FORESTRY
data in MYREGION)
```

An example of COUNT used with the same <class arithmetic expression> is

```
FOR MYREGION CALCULATE COUNT FORESTRY ACRES#
```

which produces

```
COUNT      OF FORESTRY ACRES
(<number> of occurrences of FORESTRY ACRES in MYREGION)
```

Combinations of TOTAL, PERCENT, AVERAGE, and COUNT are shown with NARIS responses in the following examples:

1) FOR MYREGION CALCULATE TOTAL, AVE FORESTRY ACRES#

   response:

```
TOTAL      AVERAGE    OF FORESTRY ACRES
<number>   <number>
```

2) FOR MYREGION CALCULATE AVERAGE, COUNT FORESTRY ACRES#

   response:

```
AVERAGE    COUNT      OF FORESTRY ACRES
<number>   <number>
```

3) FOR MYREGION CALCULATE SUM, AVE, COUNT SOIL ACRES/SUMACRES*100#

   response:

```
TOTAL      AVERAGE    COUNT        OF SOIL ACRES/SUMACRES*100
<number>   <number>   <number>
```

4) FOR MYREGION CALCULATE AVERAGE SOIL ACRES#

response:

```
AVERAGE    OF SOIL ACRES
<number>
```

3.5.2.3 CALCULATE with WHERE

A CALCULATE request may contain a WHERE <where clause> - see 3.5.2.1. This section does not include CALCULATE requests which contain both BY and WHERE.

The WHERE <where clause> in a CALCULATE request is used to instruct NARIS that the operation to be performed (e.g., TOTAL <class arithmetic expression>) is to be done only WHERE certain conditions of the data occur. For example, if one wanted to know the number of acres of SOIL NUMBER 103 which occur in the REGION he has created, the WHERE <where clause>,

...WHERE SOIL NUMBER IS 103...#

would be included in the request,

FOR MYREGION CALCULATE TOTAL SOIL ACRES...#

to produce the following CALCULATE request:

FOR MYREGION CALCULATE TOTAL SOIL ACRES WHERE NUMBER IS 103#

If the WHERE <where clause> was satisfied, NARIS would respond with

```
NUMBER EQ 103 SATISFIED
TOTAL      OF SOIL ACRES
<number>
```

If, however, the WHERE <where clause> was not satisfied (e.g., no occurrences of <data element value> "103" for SOIL NUMBER were in the REGION), NARIS would respond with

THERE WERE NO TRACTS SATISFACTORY FOR CALCULATING

In the CALCULATE request above, NUMBER was used rather than the <data element name> (SOIL NUMBER) because the <class name> (SOIL) had already been used and is understood to be a part of all remaining <data element names>.

The position of the WHERE <where clause> in a CALCULATE request is fluid, such that one may request

FOR MYREGION CALCULATE WHERE SOIL NUMBER IS 103 TOTAL ACRES#

Some examples of CALCULATE requests with a WHERE <where clause> are:

1) FOR MYREGION CALCULATE TOTAL, AVE SOIL ACRES WHERE NUMBER IS ONE OF (S103, W103, 103) AND EROSION LEQ 2#

2) FOR MYREGION CALCULATE PERCENT SOIL ACRES WHERE NUMBER IS 152#

3) FOR MYREGION CALCULATE COUNT WELL DEPTH WHERE DEPTH GTR 0#

4) FOR MYREGION CALCULATE WHERE STREAMS FLOODACRES GTR 0 TOTAL FLOODACRES#

3.5.2.4 CALCULATE with BY

BY <data element name> is a construct which may be used within a CALCULATE request (see CALCULATE definition - 3.5.2.1).

The BY <data element name> is used to structure the CALCULATE output

such that the operation being performed on the <class arithmetic expression> is presented for each value of the <data element name>.

For example, if one wanted to know the number of TOTAL acres of each SOIL NUMBER in a REGION, one could request

FOR MYREGION CALCULATE TOTAL SOIL ACRES BY NUMBER#

Let us assume that the REGION contains the following SOIL NUMBERs: 27, 56, 57, 59, 103, 148, and 152. NARIS would respond with:

```
SOIL NUMBER    TOTAL      OF SOIL ACRES
    27         <number>
    56         <number>
    57         <number>
    59         <number>
   103         <number>
   148         <number>
   152         <number>

TOTAL = <number>
```

The "TOTAL=<number>" line in the NARIS response is the total number of acres of SOIL in the REGION. This line will always appear when BY is used in a CALCULATE request.

One should note that as with the WHERE <where clause>, the BY <data element name> portion of the CALCULATE request may be placed before or after the <class arithmetic expression> and the word preceding it.

For example, one could request

FOR MYREGION CALCULATE BY LANDUSE1 CODE AVERAGE ACRES#

but <u>not</u>

FOR MYREGION CALCULATE TOTAL BY GEOWASTE TYPE ACRES#

Neither a BY <data element name> nor a WHERE <where clause> may be placed between the word(s) used to denote the CALCULATE operation (TOTAL, AVERAGE, PERCENT, COUNT) and the <class arithmetic expression>.

Some examples of CALCULATE requests using BY are:

1) FOR MYREGION CALCULATE TOTAL FORESTRY ACRES BY COVERTYPE#

2) FOR MYREGION CALCULATE AVE, PERCENT GEOSURFICIAL ACRES BY TYPE#

3) FOR MYREGION CALCULATE SUM LANDUSE1 ACRES BY CODE#

4) FOR MYREGION CALCULATE TOTAL, AVERAGE, PERCENT, COUNT LANDUSE1 ACRES BY CODE#

5) FOR MYREGION CALCULATE TOTAL SOIL ACRES BY EROSION#

3.5.2.5 CALCULATE with By and WHERE

A CALCULATE request may contain both a BY <data element name> and a WHERE <where clause>. Using both the BY and WHERE in CALCULATE allows one to:

1) find out about a select number of values of a <data element name> rather than all of them.

For example,

FOR MYREGION CALCULATE TOTAL LANDUSE1 ACRES BY CODE WHERE CODE IS ONE OF (11.1, 11.2, 11.3, 15.1, 15.2, 15.3, 18.0)#

would produce the following response

```
CODE IS ONE OF (11.1,11.2,11.3,15.1,15.2,15.3,18.0) SATISFIED
LANDUSE1 CODE       TOTAL OF LANDUSE1 ACRES
     11.1           <number>
     11.2           <number>
     11.3           <number>
```

```
15.1        <number>
15.2        <number>
15.3        <number>
18.0        <number>
```

TOTAL = <number>

2) find out about all values of a <data element name> which occur in "plots" of a specific size

For example,

FOR MYREGION CALCULATE TOTAL SOIL ACRES BY NUMBER WHERE NUMBER IS ONE OF (148, 152) AND ACRES GTR 5#

would result in (if the WHERE <where clause> was satisfied)

```
NUMBER IS ONE OF (148, 152) AND ACRES GT 5 SATISFIED

SOIL NUMBER     TOTAL OF SOIL ACRES

    148             <number>
    152             <number>

TOTAL = <number>
```

3) in terms of acreage find the values of a <data element name> which occur when other data attributes are present.

For example,

FOR MYREGION CALCULATE TOTAL SOIL ACRES BY NUMBER WHERE EROSION LEQ 1#

Thus, one would find all SOIL occurrences in the REGION which had an EROSION value less than or equal to one. The output would list the type of SOIL (SOIL NUMBER) and the total acreage for each type in the REGION WHERE the EROSION value of the occurrence was satisfactory.

The positioning of the BY <data element name> and the WHERE <where

clause> is very fluid as shown by the following CALCULATE requests:

FOR MYREGION CALCULATE TOTAL SOIL ACRES BY NUMBER WHERE NUMBER IS NOT 27 AND EROSION GTR 2#

or

FOR MYREGION CALCULATE BY SOIL NUMBER TOTAL ACRES WHERE NUMBER IS NOT 27 AND EROSION GTR 2#

or

FOR MYREGION CALCULATE WHERE SOIL NUMBER IS NOT 27 AND EROSION GTR 2 TOTAL ACRES BY NUMBER#

and so forth.

Following the word "CALCULATE" in the request, any ordering of the BY <data element name>, the WHERE <where clause>, and the

$\left\{\begin{matrix}\text{TOTAL}\\ \text{AVE}\\ \text{PERCENT}\\ \text{COUNT}\end{matrix}\right\}$ <class arithmetic expression> is permitted.

Some examples of CALCULATE requests containing BY and WHERE are:

1) FOR MYREGION CALCULATE TOTAL SOIL ACRES BY NUMBER WHERE EROSION GTR 1 AND ACRES GTR 5#

2) FOR MYREGION CALCULATE PERCENTAGE, COUNT LANDUSE1 ACRES BY CODE WHERE CODE IS NOT 95.0#

3) FOR MYREGION CALCULATE TOTAL GEOSANDGRAVEL ACRES WHERE TYPE IS BR#

4) FOR MYREGION CALCULATE TOTAL, AVE, PERCENT FORESTRY ACRES WHERE ADEQUATEMANAGEMENT EQL Y BY COVERTYPE#

5) FOR MYREGION CALCULATE TOTAL SOIL ACRES/SUMACRES*100 BY NUMBER WHERE ((ACRES/SUMACRES*100) GEQ 50)#

3.5.2.6 AND and OF in CALCULATE

"AND" and "OF" are words which may be used in a CALCULATE request to make it more Englishlike.

In Figure III-32 (see 3.5.2.1), AND may replace the comma or be used with the comma. For example, one can request

```
...TOTAL, AVE...#
    or
...TOTAL AND AVE...#
    or
...TOTAL, AND AVE...#
```

OF may be used between the word or words denoting the kind of CALCULATE being performed and the <class arithmetic expression>. For example,

```
...AVERAGE OF <class arithmetic expression>...#
                     and
...TOTAL, AVERAGE, AND PERCENT OF <class arithmetic expression>...#
```

Some examples of CALCULATE requests with "AND" and/or "OF" are:

1) FOR MYREGION CALCULATE PERCENT OF SOIL ACRES WHERE NUMBER IS 27#

2) FOR MYREGION CALCULATE TOTAL AND PERCENT OF LANDUSE1 ACRES BY CODE#

3) FOR MYREGION CALCULATE AVERAGE AND TOTAL FUTUREIMPOUNDMENT ACRES WHERE SITENUMBER EQL 1#

4) FOR MYREGION CALCULATE TOTAL, AVERAGE, PERCENT, AND COUNT OF FORESTRY ACRES BY COVERTYPE WHERE COVERTYPE IS HN#

3.5.3 MAP

MAP is the language construct used for mapping geographical areas and data attributes which occur within them. MAP requests produce shaded maps on an electrostatic printer/plotter which are then mailed from the Center for Advanced Computation to the user.

3.5.3.1 Definition

The format of the MAP request is deceptively simple.

$$\ldots\text{MAP}\quad\left\{\begin{matrix}\text{AT 1:<integer>}\\ \text{or}\\ \text{<empty>}\end{matrix}\right\}\quad\text{<tract arithmetic expression>}\ldots\#$$

Thus, each MAP request must contain a <tract arithmetic expression> and may either contain the phrase "AT 1: <integer>" or "<empty>." The phrase "AT 1: <integer>" defines the scale of the map. The <tract arithmetic expression> of the MAP request is used to define the shading of the map.

3.5.3.2 Scaling the MAP request

3.5.3.2.1 MAP AT 1: <integer>

Scale is used in MAP requests as it is used in other mapping systems. "1: <integer>" is the scale at which the map will be produced. The user must decide what integer to use.

Since there are 63,360 inches in a linear mile, a scale (or ratio) of 1:63360 would produce a map on which each inch represented a mile.

The limitation on the physical map is that it is printed within a 9.5 inch by 9.5 inch area on the printer/plotter. Thus, a MAP request containing a scale of 1:63360 would be capable of mapping a geographic area less than or equal to 9.5 miles by 9.5 miles. If the geographic area exceeded these limits when the "1:63360" scale was used, an error message would be given which would allow the user to choose another scale.

When choosing a number for scaling the map, the user should have a good estimate of the geographic dimensions of the REGION which is being mapped.

Some comments pertaining to the scale and geographic areas:

1) A section may not scaled any larger than 1:6650

2) A township may not be scaled any larger than 1:40000

If each inch of the map is to represent 10 miles, a scale of 1:633600 could be used. USGS topographic maps are scaled as noted:

7½ minute quadrangle - 1:24000

15 minute quadrangle - 1:62500

3.5.3.2.2 MAP <empty>

One may, of course, omit the "AT 1: <integer>" portion of the MAP request and let NARIS scale the map. NARIS will choose the largest possible scale (or the previously used scale, if it fits within the 9.5 inch x 9.5 inch area) which will allow the REGION to be mapped in the 9.5 inch x 9.5 inch area.

3.5.3.3 Shading of the map

Maps are produced with 8 levels of shading. To determine how the map is to be shaded, one must know the range of the data attributes (values) which are being mapped, such that new values may be calculated which fall in the range zero through seven.

NARIS maps values from ¼ ¼ sections in the following manner:

<tract arithmetic expression> value	Shading
less than 1	none
1 - less than 2	
2 - less than 3	
3 - less than 4	
4 - less than 5	
5 - less than 6	
6 - less than 7	
7 - or greater	darkest

The shadings for these values become darker as the value of the <tract arithmetic expression> becomes larger

One should form the <tract arithmetic expression> of the MAP request such that the values of the selected ¼ ¼ sections fall within the range of 0-7.

For example, the range of values for the <data element name>, SOIL ACRES, is 1-40. If one were to use SOIL ACRES as the <tract arithmetic expression>, the range of values would need to be modified from 1-40 to 0-7. Thus,

...(SOIL ACRES)/(40/7)...#

could be used as a part of the <tract arithmetic expression>.

Let us assume that "AREGION" is the name of a REGION consisting of

the contiguous tracts A, B, C, and D which contain the following data for SOIL:

	$\frac{1}{4}$ $\frac{1}{4}$ A	
NUMBER	ACRES	EROSION
103	5	0
148	10	0
103	3	1
27	15	0
152	7	2

	$\frac{1}{4}$ $\frac{1}{4}$ B	
NUMBER	ACRES	EROSION
27	6	0
353	9	0
103	17	1
344	8	0

	$\frac{1}{4}$ $\frac{1}{4}$ C	
NUMBER	ACRES	EROSION
27	20	0
103	10	0
148	5	0
152	5	0

	$\frac{1}{4}$ $\frac{1}{4}$ D	
NUMBER	ACRES	EROSION
103	31	1
152	9	0

A <tract arithmetic expression> of

...TOTAL (SOIL ACRES)/(40/7) WHERE NUMBER IS 27...#

would result in the following values being assigned to the respective $\frac{1}{4}$ $\frac{1}{4}$'s.

A	B
2.6+	1+
3.5	0
C	D

Thus, if the MAP request,

FOR AREGION MAP TOTAL SOIL ACRES/(40/7) WHERE NUMBER IS 27#

was submitted, a map would be produced which would contain each of the four lightest shades.

If one knew that the highest value for "SOIL ACRES WHERE NUMBER IS 27#" was 20, the <tract arithmetic expression> could be modified so that the value "20" would become "7". For example, either of the following requests,

MAP 2*(TOTAL(SOIL ACRES)/(40/7) WHERE NUMBER IS 27)#

or

MAP TOTAL (SOIL ACRES)/(20/7) WHERE NUMBER IS 27#

would produce values for the example $\frac{1}{4}$ $\frac{1}{4}$'s as

A	B
5+	2+
7	0
C	D

Thus, the MAP request,

MAP TOTAL SOIL ACRES/ (20/7) WHERE NUMBER IS 27#

would produce a map with shadings (from 0-7) of 0, 2, 5, and 7 which would be more distinctive than in the first request.

The following figure is a map produced by the NARIS system of a township in Lake County, Illinois depicting residential areas:

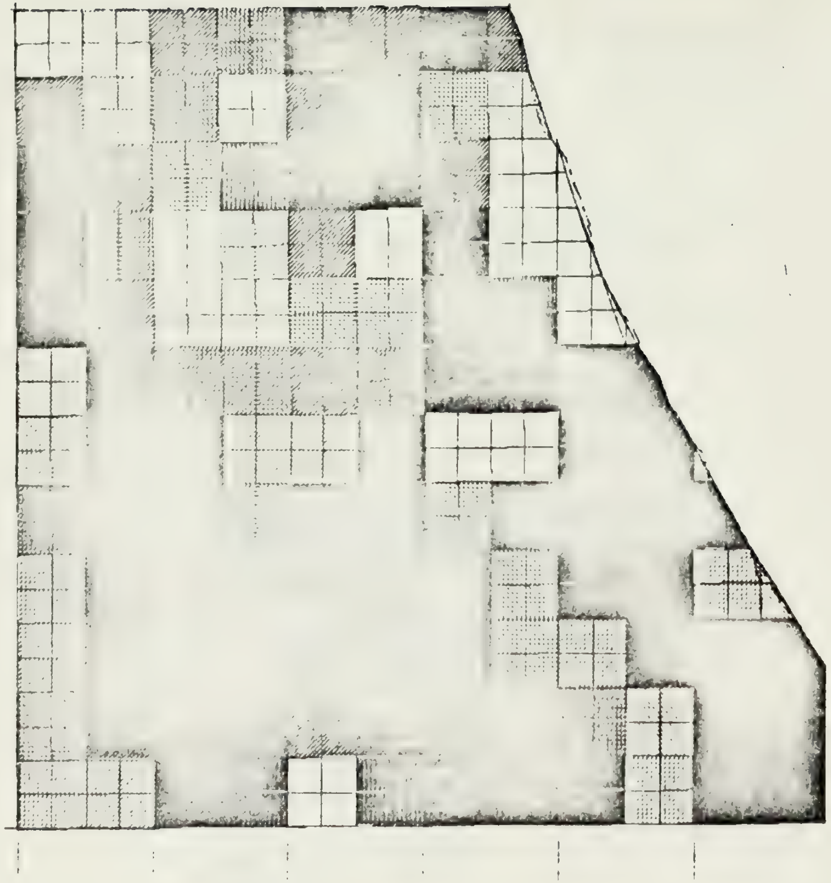


Figure III- 25. FOR T43N R12E MAP (TOTAL LANDUSE2 ACRES WHERE CODE IS ONE OF (1,2,3))/2#

The shading expression was derived by knowing that occurrences of LANDUSE2 CODE 1, 2, and 3 are very seldom greater than 14 acres. Thus, the values of LANDUSE2 ACRES were divided by 2 so that they would be expected to fall in the range of 0-7.

It should be noted that any data values which were greater than 14 acres would be treated as though they were 7 for purposes of shading. Thus, the darkest shade (7) on a map represents all data values which have been calculated to be equal to or greater than 7.

Also any data value which has been modified such that it is less than zero is considered to be zero when the map is shaded.

3.5.3.4 MAP Request Messages

Some NARIS messages which often occur when processing a MAP request are:

1) If $\frac{1}{4}$ $\frac{1}{4}$(s) were found to contain a value greater than 7, NARIS will state the following:

   <number> TRACTS WERE GIVEN A SHADING DARKER THAN THE ALLOWABLE MAXIMUM. THEY WILL BE PLOTTED AT THE ALLOWABLE MAXIMUM.

2) If $\frac{1}{4}$ $\frac{1}{4}$(s) were found to contain a value less than zero, NARIS will state

   <number> TRACTS WERE GIVEN A SHADING LIGHTER THAN THE ALLOWABLE MINIMUM. THEY WILL BE PLOTTED AT THE ALLOWABLE MINIMUM.

3) If all $\frac{1}{4}$ $\frac{1}{4}$'s to be mapped are found to have either errors in the <tract arithmetic expression> or no map data, NARIS will print

   NO POINTS SUITABLE FOR MAPPING WERE FOUND.

4) If the scale "AT 1: <integer>" was not acceptable, NARIS would respond with

MAP IS TOO LARGE TO FIT ON A 9.5 X 9.5 PLOT. A TEMPORARY SCALE HAS BEEN COMPUTED.

5) Immediately prior to drawing the map, NARIS will state:

THE SCALE IS <number>. THE MAP WILL MEASURE <number> INCHES BY <number> INCHES. DO YOU WANT IT DRAWN?

NARIS will print ":" expecting a response to the question. Four responses by the user are allowed:

1. YES# or #
in which case, a message will be printed:

DRAWING OF THE MAP HAS BEEN INITIATED.

No other message will appear until the map has been completed (in which case, NARIS will tell the user ***MAP COMPLETED (time and cost)***) unless an error occurs, in which case the following message would be printed

A FATAL ERROR OCCURRED DURING THE DRAWING OF THE MAP. REQUEST IS BEING DISCONTINUED. NO MAP WILL BE DRAWN.

2. STOP#
One may decide not to have the MAP printed and request "STOP#" in answer to the question. NARIS will respond with

NO MAP WILL BE DRAWN.

3. <integer>#
One may decide to try a new scale for the map and enter "<integer>#". The number entered would replace the integer which had been used in "AT 1: <integer>".

NARIS will respond with message 5); an attempt will be made to process the MAP request with the new scale.

4. (garbage)#
One may enter any other combination of characters which ends with a # sign. NARIS will respond with the following message:

IF YOU WISH TO PLOT THE MAP, PLEASE REPLY "YES". IF YOU WISH TO TERMINATE THIS REQUEST, REPLY "STOP". IF YOU WISH TO PLOT THE MAP WITH A DIFFERENT SCALE FACTOR, PLEASE ENTER THE NEW SCALE AS A SINGLE INTEGER NUMBER.

3.5.3.5 Examples of MAP Requests

1) FOR MARENGOTWP WHERE SOIL NUMBER EQL 27 AND ACRES GEQ 5 MAP AT 1:40000 7#

This request will produce a map at the darkest shading (7) in each $\frac{1}{4}$ $\frac{1}{4}$ of Marengo township which satisfies the WHERE <where clause>.

2) ABBREVIATION SOILINMAP. IS 2* (TOTAL SOIL ACRES/(40/7) WHERE NUMBER IS 103)#

ABBREVIATION NOFOREST. IS (-100*(TOTAL FORESTRY SUMACRES WHERE SUMACRES GTR 0))#

FOR MARENGOTWP MAP AT 1:40000 ((SOILINMAP.)+NOFOREST.)#

This map would reflect the acreage of SOIL NUMBER 103 in $\frac{1}{4}$ $\frac{1}{4}$'s which do not contain FORESTRY.

3) See section 2.3, Introductory Problem Using the Language.

3.5.4 OUTPUT

The OUTPUT language construct provides users the capability of creating disk files on the B6700 computer which contain the results of the numerical calculations which have been performed on NARIS <data element values> on a $\frac{1}{4}\,\frac{1}{4}$ section by $\frac{1}{4}\,\frac{1}{4}$ section basis. The user may then

- copy the file to a tape; and/or
- write a program which uses the file as data - this may be done for any computer on which the tape can be read.

The OUTPUT language construct is designed to be used by those who are familiar with computer files, have access to a computer, and are capable of performing operations on the content of the file.

The user of the OUTPUT language contruct is referred to the B6700 documentation on file naming, file usage and programming languages or to the documentation of other systems such as MONICA (programmed at the Center for Advanced Computation) which accept NARIS OUTPUT files.

3.5.4.1 Definition

An OUTPUT request has the following format

```
    ( OUTPUT ON FILE "<file name>"    )   ,-------------------------------,
... (            or                   )  v <tract arithmetic expression>  ...#
    ( ON FILE "<file name>" OUTPUT    )
```

The arrow is used to signify that a comma must be used as a separator between <tract arithmetic expressions>. Up to three <tract arithmetic

expressions> may be used in an OUTPUT request, e.g.,

```
...OUTPUT ON FILE "<file name>"  <tract arithmetic expression>,
<tract arithmetic expression>, <tract arithmetic expression>...#
```

NARIS will create the disk file from OUTPUT such that the name of the disk file is of the format:

NARIS/<user number>/<file name>

A user may obtain his <user number> from the NARIS User Liaison at the Center for Advanced Computation.

One must note that disk files created by the user will automatically be SAVEd.

If no <existing region> is specified with an OUTPUT request, the last REGION which was used in a request (the FOR-CLAUSE-REGION) will be used.

3.5.4.2 OUTPUT FILE STRUCTURE

Each record in the file is 84 bytes (14 6-byte B6700 words) long. Only the first 72 bytes of each record contain information. There are 30 records to a block (and thus 2520 bytes or 420 B6700 words).

The first record contains, in EBCDIC characters, a FORTRAN format that defines the fields of the remaining records. The contents of the first record are:

%(I1,1X,I1,1X,I2,1X,I3,1X,I3,1X,E19.12,E19.12,E19.12)

Each following record contains EBCDIC characters giving tract indentification information and values for the <tract arithmetic expressions> according to the format specified in the first record.

The five integer fields will respectively contain the following tract identification information processed from the SURVEY Data Class:

Byte number in record		Possible values and meaning
1	QUARTER-QUARTER SECTION equals	0 meaning "NEQ" 1 meaning "NWQ" 2 meaning "SEQ" 3 meaning "SWQ"
3	QUARTER SECTION equals	0 meaning "NEQ" 1 meaning "NWQ" 2 meaning "SEQ" 3 meaning "SWQ"
5-6	SECTION	equals section number
8-10	TOWNSHIP	equals township number if north 64 + township number if south
12-14	RANGE	equals range number if east 64 + range number if west

The three "E" fields will contain the values of the <tract arithmetic expressions> in the order specified within the OUTPUT statement. If only one <tract arithmetic expression> is specified, zero will be placed in the second and third "E" fields. Similarly, if two <tract arithmetic expressions> are specified, zero will be placed in the third "E" field.

3.5.4.3 Example of an OUTPUT Request

The following OUTPUT request

```
FOR SEC 31 T44N R5E OUTPUT ON FILE "MYUSERFILE" FUNCTION  [(0,100
(50,25) (100,100)]  ((TOTAL FORESTRY ACRES/40)*100,10#
```

will create a disk file with the name "NARIS/<your user number>/MYUSERFILE."

The file will contain the following data:

```
RECORD 0:
%(I1,1X,I1,1X,I2,1X,I3,1X,I3,1X,E19.12,E19.12,E19.12)
RECORD 1:
0 0 31  44   5  1.000000000000E 02 1.000000000000E 01 0.000000000000E 00
RECORD 2:
1 0 31  44   5  1.000000000000E 02 1.000000000000E 01 0.000000000000E 00
RECORD 3:
2 0 31  44   5  1.000000000000E 02 1.000000000000E 01 0.000000000000E 00
RECORD 4:
3 0 31  44   5  7.800000000000E 01 1.000000000000E o1 0.000000000000E 00
                                  .
                                  .
                                  .
```

3.6 ON

The means by which output is produced for the four language constructs --MAP, OUTPUT, TABULATE, and CALCULATE, has been defined in the previous section (3.5).

MAP produces a shaded map at the Center for Advanced Computation which is mailed to the user.

OUTPUT produces a disk file on the B6700 computer with the name of "NARIS/<user number>/<file name>."

TABULATE and CALCULATE produce output on the user's terminal. By using "ON", however, the output of CALCULATE and TABULATE may be directed to a device other than the terminal.

3.6.1 TABULATE ON PRINTER

3.6.1.1 Definition

ON is used with a TABULATE request with the word, PRINTER, as follows:

```
ON PRINTER FOR <existing region> TABULATE...#
FOR <existing region> ON PRINTER TABULATE...#
ON PRINTER TABULATE FOR <existing region>...#
TABULATE ON PRINTER FOR <existing region>...#
TABULATE FOR <existing region> ON PRINTER...#
```

The only restriction regarding the ordering of the phrases FOR <existing region>, TABULATE, and ON PRINTER is that they must precede the "object" of the TABULATE (that which is to be TABULATEd) - the exception to this

is FOR <existing region> which may follow the object of the TABULATE.

TABULATE requests with ON PRINTER will cause NARIS to put the output on a printer listing. The listing is then mailed to the user from the computer center.

3.6.1.2 Examples of TABULATing ON PRINTER

```
1)  ON PRINTER FOR MYREGION TABULATE WHERE LANDUSE1 CODE IS 33.0#
2)  FOR MYREGION TABULATE ON PRINTER SOIL, LANDUSE1, LANDUSE2,
    WHERE FUTUREIMPOUNDMENT ACRES GTR 0#
3)  FOR MYREGION TABULATE ON PRINTER TOTRSCS.#
4)  FOR MYREGION TABULATE ON PRINTER SOIL NUMBER, (SOIL ACRES/
    SUMACRES)*100 WHERE SLOPE IS NOT ONE OF (A,B), FORESTRY#
```

3.6.2 CALCULATE ON

ON may be used with CALCULATE to specify that the CALCULATE output be produced ON PRINTER or ON FILE <file name>.

3.6.2.1 CALCULATE ON PRINTER

One may specify that one's CALCULATE output be printed on a computer print out listing. The output (listing) is mailed to the user.

The format of requesting that one's CALCULATE output be in the form of a computer listing is exactly the same as that for TABULATE ON PRINTER (section 3.6.1.1) with the exception that the word, CALCULATE, replaces the word TABULATE.

3.6.2.1.1 Examples

```
1)  ON PRINTER FOR MYREGION CALCULATE TOTAL AND PERCENT OF SOIL ACRES
    BY NUMBER#
```

2) ON PRINTER FOR MYREGION CALCULATE TOTAL, AVERAGE AND COUNT OF FORESTRY ACRES WHERE COVERTYPE EQL HN#

3) FOR MYREGION CALCULATE ON PRINTER TOTAL AND COUNT OF FUTUREIMPOUNDMENT ACRES BY SITENUMBER WHERE ACRES GTR 0#

3,6.2.2 CALCULATE ON FILE "<file name>"

CALCULATE output may be directed to a disk file on the B6700 computer - the disk file name being designated by the user. The format for CALCULATing ON FILE "<file name>" is

```
...FOR <existing region> CALCULATE ON FILE "<file name>"...#
```

Again, the positioning of the words and phrases (FOR <existing region>, CALCULATE, and ON FILE "<file name>") is flexible. such that any ordering is acceptable. The object of the CALCULATE must, however, be placed at the end of the request.

As with OUTPUT, the disk file created by CALCULATE ON FILE "<file name>" will have the name "NARIS/<user number>/<file name>."

The file will contain a FORTRAN FORMAT of the file in record zero. Records 1 to <end-of-file> will contain CALCULATE results. The results of the CALCULATE will differ from the CALCULATE ON PRINTER in the following way:

- If a BY is used in the CALCULATE request, all <data element values> (including zero) will be included in records of the disk file.

Thus, it is very possible that many records of the disk file will contain values of zero.

3.6.2.2.1 Examples

1) CALCULATE FOR MYREGION ON FILE "MYFILE" TOTAL SOIL ACRES/SUMACRES 100 BY NUMBER#

2) FOR MYREGION CALCULATE ON FILE "MYFILE" TOTAL GEOSURFICIAL ACRES BY TYPE#

3) ON FILE "MYFILE" FOR MYREGION CALCULATE TOTAL AND AVE OF LANDUSE2 ACRES BY CODE WHERE CODE IS ONE OF (10,11)#

3.6.3 Table of using ON

ON may be used with language constructs as follows:

language constructs	ON PRINTER	ON FILE "<file name>"	If ON is not used, output will be generated to the TI terminal, except for MAP
MAP			X
TABULATE	X		X
CALCULATE	X	X	X
OUTPUT		X	

Figure III- 26. Language construct table with ON.

IV The IRIS I System

Much of the presently available resource data in Illinois has been collected on a ¼ section basis. Because NARIS can only be used with ¼ ¼ section data, the Center for Advanced Computation has developed IRIS I (Illinois Resources Information System), a NARIS counterpart which uses ¼ section data.

IRIS I and NARIS are similar systems both in philosophy and implementation; their language constructs are practically identical. The two systems, however, are strictly independent from each other, and information which is SAVEd under NARIS, such as REGIONS, ABBREVIATIONS, and FUNCTIONS, cannot be accessed by IRIS I, and vice-versa. (In the case of REGIONS, for instance, even though a NARIS REGION and an IRIS I REGION might describe the same geographic area, they contain internal references to their respective data bases, and therefore cannot be used interchangeably.)

These restrictions will be removed in subsequent versions of IRIS. In particular, IRIS II will allow access to several Data Bases with a variety of geographic units.

4.1 The IRIS I Data Base

4.1.1 Differences from the NARIS Data Base

The IRIS I data base is quite similar in organization to the NARIS data base. In IRIS I, however, the basic unit for storing data into the

data base is the ¼ section (160-acre tracts of land). The implications of this difference in data resolution are:

- IRIS I data and NARIS data are incompatible in that they reference different geographic units.
- Not all Data Classes are the same in both systems. (See IRIS I Data Guide).
- Formats for IRIS I data coding and data base insertion are slightly modified from the NARIS format to accommodate ¼ section resolution. (See IRIS I addenda to the Data Insertion and Data Coding Manuals).
- Any given Data Class of NARIS ¼ ¼ section data may be aggregated into IRIS ¼ section data; however, the data must be re-coded and inserted into the IRIS I data base.
- The numbering scheme for ¼ sections and sections within townships is the same as that used in NARIS (see figure I-1, page I-6).

4.1.2 IRIS I Data Base Contents

Some of the projected categories of data to be included in the IRIS I data base are:

- Interim Interpretative Resource Data (NIPC)
- Land Use Data (NIPC)
- Employment Data (ISEC/NIPC)
- Housing and Demographic Data (Census)

- Growth Monitoring Data (NIPC)
- Forecasts of Population, Employment and Land Use (NIPC)

4.2 Getting Into the IRIS I System

Set up the TI terminal and get a connection to the computer as you would with NARIS. When the system responds with

```
***THIS IS NARIS/IRIS, WHO ARE YOU?***
:
```

You may use NARIS by simply typing your last name, followed by "#" and a carriage return. If you wish to use IRIS I, type your last name followed by "$IRIS#" and a carriage return.

The NARIS system maintains a NARIS/IRIS option for each user, whose default value is usually "NARIS". In this case you would log in as described above. If you are a regular IRIS I user, you may arrange to have your default value changed to "IRIS", in which case you would simply type your name, "#" and a carriage return when using IRIS I, or your name, "$NARIS#" and a carriage return when using NARIS.

For example,

A. If the default option is NARIS

When using NARIS:

```
<user's last name>#

        or
```

```
        <user's last name>$NARIS
When using IRIS I:
        <user's last name>$IRIS
```

B. If the default option is IRIS

When using NARIS:

```
        <user's last name>$NARIS
```

When using IRIS I:

```
        <user's last name>#
             or
        <user's last name>$IRIS
```

The remainder of the log-on procedure, and the entire log-off procedure are the same as in NARIS (see section 2).

4.3 The IRIS I Language

The syntax of the IRIS I language is identical to that of the NARIS language, with the exception of <tract specification> (see 4.3.1.1.2)

4.3.1 User Information which is Saved

4.3.1.1 REGIONS

A REGION in IRIS I is a list of ¼ sections representing a specific

geographic area in the IRIS I data base; whereas its NARIS counterpart consists of a list of ¼ ¼ sections in the NARIS data base. Hence, IRIS I and NARIS REGIONS cannot be used interchangeably, even if they represent the same geographic area. Their definitions in the language differ only insofar as <tract specification> is concerned.

4.3.1.1.1 FOR-CLAUSE REGIONS

FOR-CLAUSE REGIONS are identical in concept in both IRIS I and NARIS. The only difference, as noted in 4.3.1.1, is the <tract specification> portion of the language.

4.3.1.1.2 TRACT SPECIFICATION

Since the basic geographic unit for IRIS I is the ¼ section, specifications of ¼ ¼ sections are illegal, and as a result IRIS I <tract specifications> are less complex than those for NARIS. Specifically, parentheses are not allowed, and combinations of tracts within sections are usually shorter.

For instance:

LEGAL	ILLEGAL (But legal in NARIS)
NEQ SEC 17 T44NR5E	NEQ NEQ SEC 2 T41NR8E
SH, NWQ SEC 2, 11 T42-44NR4E	(SH, NWQ) SWQ SEC 6, 32 T43NR10W

SEC 1-12, EH SEC 18 T38UR10E   SH, SEQ NWQ, (WH, SEQ) NEQ SEC 1
T4ONR11E

4.3.1.1.3 REGION request with <where clause>

This construct is identical to the NARIS construct. Its meaning differs only in that <where clauses> are processed by ¼ sections within the specified area, rather than by ¼ ¼ sections.

4.3.1.1.4 Combinations of pre-existing REGIONS

All methods which are valid in NARIS for combining REGIONS are also valid in IRIS I.

4.3.1.2 ABBREVIATIONS

ABBREVIATIONS may be used in IRIS I in the same manner as they are used in NARIS. However, ABBREVIATIONS created under NARIS cannot be used under IRIS I, and vice-versa.

4.3.1.3 FUNCTIONS

As with ABBREVIATIONS, FUNCTIONS are identical in both systems, and can only be used when using the system under which they were created.

4.3.1.4 PUBLIC/PRIVATE

As in NARIS, each IRIS I REGION, ABBREVIATION and FUNCTION which the user created is considered by the system to be PRIVATE unless the user specifies it to be PUBLIC or SEMIPUBLIC.

4.3.2 Language Constructs which do not access the Data Base

This group of constructs, namely, SAVE, MAKE, FORGET, LIST and WHAT IS (ARE) is identical to the corresponding NARIS constructs. Note that IRIS I requests will only operate on IRIS I information, and hence "LIST REGIONS", for instance, will not provide any information about REGIONS created under NARIS.

4.3.3 Arithmetic Expressions

4.3.3.1 Class Arithmetic Expressions

There is no difference between NARIS and IRIS I <class arithmetic expressions>.

4.3.3.2 Tract Arithmetic Expressions

Since the geographic resolution is different in each system, <tract arithmetic expressions> are processed according to each system's basic geographic unit (¼ sections in IRIS I and ¼ ¼ sections in NARIS).

4.3.4 WHERE <where clause>

<where clauses> serve the purpose of selecting only those tracts within a specified area which meet the conditions defined in the <where clause>. IRIS I <where clauses> apply to entire ¼ sections at a time, unlike NARIS <where clauses>, which apply to ¼ ¼ sections.

4.3.5 Language Constructs which access the Data Base

4.3.5.1 TABULATE

TABULATE functions the same way as it does under NARIS. Its output differs in that the tract identification headers pertain to ¼ sections (e.g., NEQ SEC 1 T44NR5E:).

4.3.5.2 CALCULATE

CALCULATE requests are also identical in both systems. Internally, however, CALCULATE aggregates data by ¼ seciton rather than by ¼ ¼ section.

4.3.5.3 MAP

IRIS I maps are produced with ¼ section resolution, in contrast with NARIS maps, which select shadings by the ¼ ¼ section. In all other respects, the mapping procedures are identical in both systems.

4.3.5.4 OUTPUT

The format for OUTPUT requests is the same for both IRIS I and NARIS. The disk file which is created, however, will bear the name

IRISUSER/U<user number>/<file name>

instead of the prescribed NARIS title.

In addition, the tract identification information for each record created will have the following format:

Byte number	SURVEY Data element	Possible values	meaning
1	QUARTER-SECTION	0	NEQ
		1	NWQ
		2	SEQ
		3	SWQ
3-4	SECTION	section number	
6-8	TOWNSHIP	township number	north township
		64 + township number	south township
		128 + township number	upper indian boundary township
		256 + township number	lower indian boundary township
10-12	RANGE	range number	east range
		64 + range number	west range

APPENDIX A

Glossary of NARIS Terms

This Glossary is used to define words and phrases which have a special meaning as used in the NARIS User Manual. **The Index of the manual references** a more comprehensive list of terms and phrases.

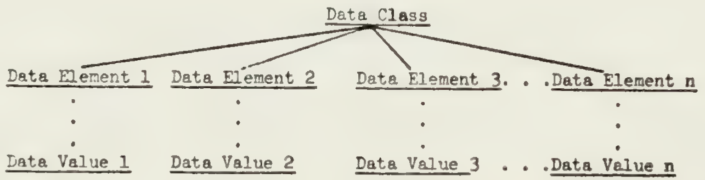
<class name>- <class name> is the name of a Data Class. For example, some <class names> are GEOSANDGRAVEL, SOIL, FORESTRY, PLANTATION, IMPOUNDMENT, and WELL.

Computer Listing or Print out - The ON PRINTER phrase may be used to produce computer listings. The language constructs which may be used with ON PRINTER may be found in 3.6.1, TABULATE ON PRINTER, and 3.6.2.1, CACULATE ON PRINTER. The listing or print out will appear on the printer at UCSD. An example of a listing is shown in Figure II-2 on page II-16.

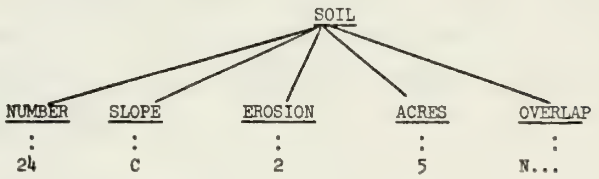
Data Base - The NARIS data base consists of all Data Class information stored by geographic location (¼ ¼ section). See 1.3, The Data Base.

Data Class - A Data Class refers to a specific type of natural resource or socio-economic information. Each Data Class comprises more specific Data Elements. The values of the Data Elements are the attributes of the ¼ ¼ sections. For further information see section 1.3 of this manual.

Data Class Occurrence - Each occurrence of a Data Class in a ¼ ¼ section is a Data Class occurrence. For example, if there were five plots of soil in a tract there would be five occurences of the SOIL Data Class. Each Data Class occurrence consists of all Data Elements of the class and their attributes for the tract. A Data Class occurrence is shown in the following diagram:



Thus, from page I-9 (figure I-2), **the 5 acre soil plot located at the north edge of the soil map would have the following structure within the data base as a SOIL data occurrence:**



For further information see pages II-15 and II-16.

Data Element - A Data Element is a component of a Data Class and comprises Data Values or attributes which define the Data Class in a ¼ ¼ section.

<data element name>- <data element name> is the name of a Data Element. The <class name> portion of a <data element name> may, under certain circumstances, be omitted. Some <data element names> of the Data Class FORESTRY are FORESTRY STANDNB, FORESTRY DATE, FORESTRY ACRES, FORESTRY COVERTYPE, and FORESTRY OVERLAP.

<data element value> - <data element value> is the value of a Data Element in a Data Class occurrence. For example, the <data element value> of a data occurrence of SOIL ACRES might be 12.

Data Occurrence - see Data Class Occurrence.

Data Value - see <data element value> .

Disk File - A disk is a magnetic data storage device. Collections of data which are physically stored on a magnetic disk are called disk files. The language constructs, CALCULATE and OUTPUT, may be used to create B6700 disk files. See section 3.6.2.1, CALCULATE ON FILE "<file name>", and section 3.5.4, OUTPUT.

interactive - Characterized by a question - answer dialogue.

Occurrence - see Data Class Occurrence.

reserved word - A word which may not be used as a <region name>, <abbreviation name>, or <function name>.

TI Terminal - The TI (Texas Instrument) terminal is the hardcopy typing console which is the user's means of connecting to the NARIS system via telephone. The telephone connection is made from the TI terminal through several other computers to the NARIS host computer in San Diego. More information regarding the TI terminal is presented in section 2.1, Getting In and Out of the NARIS System.

¼ ¼ *section or tract* - A tract, or ¼ ¼ section, is the basic unit of storing data in the NARIS data base. A ¼ ¼ section comprises 40 acres of land (due to surveying irregularities, not all tracts in the State of Illinois are exactly 40 acres in size; however, NARIS does have the capability of dealing with non-40-acre tracts). The information available for each ¼ ¼ in the data base is accessible by its geographic identification label which is derived from the legally established Rectangular Survey System. For further information see section 1.3, The Data Base, in the manual.

Appendix B. NARIS Data Guide

This guide is intended as a brief data reference for users of the NARIS system. It is a condensation of information which may be obtained in detail by using the WHAT IS and LIST requests (see section 3.2 of the NARIS User Manual).

B.1 Sources of Data

The data available to NARIS users is collected and published by several federal, state, and regional government agencies with programs in Illinois. The information comes from maps and other formal documents obtained through the cooperation of the following agencies:

State

- Illinois Department of Agriculture
  - →Division of Soil and Water Conservation
- Illinois Department of Conservation
  - →Division of Fisheries
  - →Division of Forestry
  - →Division of Wildlife Resources
- Illinois Department of Registration and Education
  - →Illinois State Geological Survey
  - →Illinois State Water Survey
- Illinois Environmental Protection Agency

Federal

- U.S. Department of Agriculture, Soil Conservation Service
- U.S. Department of Interior, Geological Survey

Regional

- DuPage County Regional Planning Commission
- Kane County Regional Planning Commission
- Lake County Regional Planning Commission
- McHenry County Regional Planning Commission
- Northeastern Illinois Planning Commission
- The Morton Arboretum

B.2 Data Encoding

The process of encoding data from the above agencies into the NARIS data base is described in detail in the NARIS Data Coding Manual. This involved decisions regarding geographic referencing, descriptions of Data Elements and values, and the development of data coding forms in cooperation with the data collection agencies. The Northeast Illinois Natural Resource Service Center played the dominant role in this coordination. The Northeastern Illinois Planning Commission and the Center for Advanced Computation also coordinated data encoding for some Data Classes.

B.3 Column Descriptions of the Following Pages

Data Element Name - the <data element names> within the Data Class are listed. The <class name> is placed at the head of the column and is intended to be used in conjunction with the words which appear below it. For example, on page B-5, STANDNB is to be read as FORESTRY STANDNB.

AE (Arithmetic Expression) - an X is placed in this column if the Data Element is a numeric Data Element which can be used within an arithmetic expression.

Values - Wherever practical all of the <data element values> are listed. For some Data Elements, such as SOIL NUMBER or LANDUSE1 CODE, the <data element values> are too numerous to list. In these cases, ellipses (...) are used to indicate that the list is not complete. A LIST <data element name> VALUES# request will show all the <data element values>. Some of the <data element values> listed contain special characters. These values must be enclosed in quotes when accessed from the terminal, e.g., GEOSURFICIAL TYPE "6/7". If the <data element value> desired is a blank, two consecutive quotes must be used, e.g., WHAT IS FORESTRY BURNT ""#. Where <data element values> are sequential, only the first and last values are indicated (e.g., FORESTRY ACRES shows 0-45, meaning 0 through 45 inclusive, and SOIL SLOPE shows A-G, meaning the letters A through G inclusive).

Notes - this column contains brief comments describing the Data Elements of the Data Class.

The following pages present a description of the contents of each NARIS Data Class: the <data element names>, their values, and a general description.

One should note that although some <data element names> have been hyphenated in these pages (e.g., COVER-OVER40 on page B-5), no <data element name> may be hyphenated when used in a NARIS request!

FORESTRY

Native woody vegetation of at least one acre in size outside corporate city limts.

Data Element Name	AE	Values	Notes
FORESTRY			
STANDNB	X	0-15	Distinguishes stands within the ¼ ¼
ACRES	X	0-45	Acreage of this stand
LOCATION		X or "," (16 characters)	Location of this stand within the ¼ ¼
OVERLAP		X or "," (4 characters)	Direction(s) this stand overlaps into adjoining ¼ ¼'s
COVER-OVER40	X	0-45	Number of acres with over 40% crown cover
COVER-UNDER40	X	0-45	Number of acres with under 40% crown cover
DATE	X	60-75	Year of last on-site inspection
SUMACRES	X	0-45	Total acreage of forestry in the ¼ ¼
COVERTYPE		HN,CS,OS,...	Most common tree species
SIZECLASS		P,T,S,L,...	Dominant size class
DENSITY	X	0-999	Average basal area per acre in sq. feet
ADEQUATE-MANAGEMENT		Y,N," "	Is the stand adequately managed?
CUT		Y,N," "	In need of harvest?
BURNT		Y,N," "	Obvious history of burning?
GRAZED		Y,N," "	Domestic livestock present?
INSECT		Y,N," "	Insect problems?

(cont.)

DISEASE	Y,N, " "	Obvious evidence of a disease which is a hazard to the stand?
COMPETITION	Y,N, " "	Undesirable trees present?
URBAN	Y,N, " "	Subdivided for homesites?
OWNERSHIP	A,S,O,F,...	Private-absentee, State, Private, Federal,...

FUTUREIMPOUNDMENT

Potential Reservoir sites.

Data Element			
Name	AE	Values	Notes
FUTUREIMPOUNDMENT			
DATE	X	60-75	Year study was published
LOCATION		X or "'" (16 characters)	Location of the impoundment within the ¼ ¼
ACRES	X	0-40	Acreage of the impoundment
OVERLAP		X or "'" (4 characters)	Direction(s) the impoundment overlaps into adjoining ¼ ¼s
TYPECONSTRUCTION		DU,DA," "	Dugout (excavated) or Dam
WATERSOURCE		ST,HW,RU," "	Stream, High water table, Runoff
MATERIALS		G,Y,R," "	Indicates availability of construction materials
ELEVATION	X	0-2047	Sea level to water level when full (in feet)
DEPTH	X	0-999	In feet
FEASIBILITY		Y,N," "	Geologically feasible?
PUBLICATION		EGN25,...	Publication source of information
SITENUMBER	X	0-1023	Each potential reservoir has a unique site number

GEOCONSTRUCT

Interpretation of the geologic limitations on construction.

Data Element Name	AE	Values	Notes
GEOCONSTRUCT			
TYPE		G2,Y2,R1,...	Estimate of the amount of limitation
ACRES	X	10,20,30,40	Acreage of this type - 10 acre resolution
DATE	X	60-75	Year of published survey
SUMACRES	X	40	Total acreage of GEOCONSTRUCT data in the ¼ ¼.

GEOSANDGRAVEL

Sand and gravel mineral resource interpretation.

Data Element Name	AE	Values	Notes
GEOSANDGRAVEL			
TYPE		G1,Y2,Y3,...	Estimate of the quality of sand and gravel resources
ACRES	X	10,20,30,40	Acreage of this type - 10 acre resolution
DATE	X	60-75	Year of published survey
SUMACRES	X	40	Total acreage of GEOSANDGRAVEL data in the ¼ ¼.

GEOSURFICIAL

Surficial deposits

Data Element Name	AE	Values	Notes
GEOSURFICIAL			
TYPE		5D,"6/7",15,...	Surface geological materials
ACRES	X	10,20,30,40	Acreage of this type - 10 acre resolution
DATE	X	60-75	Year of published survey
SUMACRES	X	40	Total acreage of GEOSURFICIAL data in the ¼ ¼.

GEOWASTE

Waste disposal capability interpretation

Data Element Name	AE	Values	Notes
GEOWASTE			
TYPE		G3,Y1,R3,...	Estimate of capability for waste disposal
ACRES	X	10,20,30,40	Acreage of this type - 10 acre resolution
DATE	X	60-75	Year of published survey
SUMACRES	X	40	Total acreage of GEOWASTE data in the ¼ ¼.

GEOWATER

Groundwater potential interpretation

Data Element Name	AE	Values	Notes
GEOWATER			
TYPE		G1,G2,Y1,...	Estimate of Groundwater Potential
ACRES	X	10,20,30,40	Acreage of this type - 10 acre resolution
DATE	X	60-75	Year of published survey
SUMACRES	X	40	Total acreage of GEOWATER data in the ¼ ¼.

IMPOUNDMENT

A permanent body of water one-third acre or larger

Data Element Name	AE	Values	Notes
IMPOUNDMENT			
DATE	X	60-75	Year data was last updated
LOCATION		X or "," (16 characters)	Location of the impoundment within the ¼ ¼
TYPE		N,M," "	Natural or Manmade
OVERLAP		X or "," (4 characters)	Direction(s) this impoundment overlaps into adjoining ¼ ¼'s
CONSTRUCTION		F,D, " "	Filled (dammed) or Dugout (excavated)
ACRES	X	0-40.0	Resolution to .1 acre
YEARCON-STRUCTED	X	0-75	Applicable only if man-made
OWNERSHIP		PR,PU," "	Private or Public
DEPTH	X	0-999	Maximum depth from surface to bottom in feet
FISHSOURCE		N,S," "	Natural or stocked
YEARSTOCKED	X	0-75	Applicable only if stocked
RECREATION		X or ","	Principal use(s) of this impoundment (X denotes a use)
MUNICIPAL		X or ","	
INDUSTRIAL		X or ","	
AGRICULTURAL		X or ","	
IRRIGATION		X or ","	
FLOODCONTROL		X or ","	
WASTEWATER		X or ","	

(cont.)

PUBLICATION	GESU,SCS,SWCD,...	Indicates the publication source
SPECIES	A-O," "	Code for dominant species of fish

LANDUSE1

Land use survey provided by the Northeast Illinois Natural Resource Service Center, Lisle, Illinois

Data Element Name	AE	Values	Notes
LANDUSE1			
CODE		07.1,46.0,84.2,...	McHenry County land use code numbering scheme
ACRES	X	0.0-45.0	Resolution to .1 acre
COUNT	X	0-99	Number of establishments of this code in this ¼ ¼
ROADFEET	X	0-9999	Linear feet of improved road when code =07.1 or 08.1
DATE	X	60-75	The year the study was completed

LANDUSE2

Land use survey provided by Northeastern Illinois Planning Commission

Data was collected with ¼ section resolution

Data Element Name	AE	Values	Notes
LANDUSE2			
DATE	X	60-75	Year survey was made
SUMACRES	X	0-45.00	Total acreage accounted for in this ¼ ¼
CODE		1,2,...,18	Land use code
ACRES	X	0-45.00	Resolution of .01 acre (¼ section acres divided by 4)
INCORPORATED		I,U," "	Incorporated, unincorporated
COUNT	X	0-999	Number of establishments of this code in this ¼ ¼
BOUNDARY		Y,N," "	Does a political boundary subdivide this ¼ ¼?

MAPDATA

Encoded Lambert coordinates. This class is automatically used by the MAP request

Data Element Name	AE	Values	Notes
MAPDATA			
X	X	0-65535	East-West coordinate divided by 64
Y	X	0-65535	North-South coordinate divide by 64

PLANTATION

Planted vegetation

Data Element Name	AE	Values	Notes
PLANTATION			
STANDNB	X	0-15	Distinguishes stands within a ¼ ¼
ACRES	X	0-45	Acreage of the stand
LOCATION		X or "'" (16 characters)	Location of stand within the ¼ ¼
OVERLAP		X or "'" (4 characters)	Direction(s) in which stand overlaps into adjoining ¼ ¼'s
DATE	X	60-75	Date of on-site inspection
SUMACRES	X	0-45	Total acreage of planted vegetation in this ¼ ¼
ADEQUATE-MANAGEMENT		Y,N," "	Is the stand adequately managed?
CUT		Y,N," "	In need of harvest?
GRAZED		Y,N," "	Domestic livestock present?
BURNT		Y,N," "	History of burning?
INSECT		Y,N," "	Insect problems?
DISEASE		Y,N," "	Obvious evidence of disease which is a hazard to the stand.
COMPETITION		Y,N," "	Undesirable trees present?
URBAN		Y,N," "	Subdivided for homesites?
OWNERSHIP		A,S,O,F,...	Absentee-private, state, private, federal,...
TYPE		N,O," "	Nursery or orchard

YEARPLANTED	X	0-75	Year of planting
SURVIVAL	X	0-95	Percentage of survival to nearest 5%
SPECIES1		A-P,R-Y," "	Species of vegetation
SPECIES2		A-P,R-Y," "	"
SPECIES3		A-P,R-Y," "	"
SPECIES4		A-P,R-Y," "	"
SPECIES5		A-P,R-Y," "	"

SOIL

Soil surface to a depth of five feet.

Data Element Name	AE	Values	Notes
SOIL			
NUMBER		27,W103,347,...	Type of soil on this plot - SCS numbering scheme
SLOPE		A-G," "	A or " " is level, G is steep
SUMACRES	X	1-45	Total acreage of the tract - 1 acre resolution
EROSION	X	0-3	Zero means no erosion
ACRES	X	1-45	Acreage of the plot
OVERLAP		X or "'" (4 characters)	Direction(s) this plot overlaps into adjoining ¼ ¼'s
DATE	X	60-75	Year of latest release of correlated soil survey by SCS

STREAMS

Intermittent and Perennial watercourses.

Data Element Name	AE	Values	Notes
STREAMS			
DATESTREAM-MAP	X	0-75	USGS quadrangle map publication date
DATEFLOOD-PLAINMAP	X	0-75	USGS/NIPC map publication date
ENTRANCE		NE,SW,SC,...	Northeast, Southwest, Southcentral,...
EXIT		NE,SW,SC,...	"
OUTLET	X	0-999	Distance in miles to outlet of stream
FLOODYEAR	X	0-75	Year of greatest flood
FLOODACRES	X	0-40	Acreage flooded during greatest flood
FLOODLOCATION		X or "'" (16 characters)	Flooded area during greatest flood
OVERLAP		X or "'" (4 characters)	Direction(s) in which flood overlaps into adjoining ¼ ¼'s
STATION		X or "'"	Gauging station?
NUMBER		05540050, 05551620,...	USGS gauging station number
NAME		1,2,3,4,...	The name of the stream
FLOW		P,I," "	Perennial or intermittent
SOURCE		USGS,NIPC,ISGS,...	Source of the information
QUALITY		A-H," "	EPA water quality
MAPNUMBER		HA89,HA204,...	Map number of USGS/NIPC map

SURVEY

Geographic location in survey notation. This class is automatically used by the TABULATE request

Data Element Name	AE	Values	Notes
SURVEY			
QQUARTER		NEQ,SEQ,NWQ,SWQ	The ¼ ¼ within the ¼
QUARTER		NEQ,SFQ,NWQ,SWQ	The ¼ within the section
SEC		"SEC 1","SEC 2", ..."SEC 36"	The section within the township
TOWNSHIP		T43N,T44N,..	Township number
RANGE		R09E,R10E,..	Range number

WATERSHED

Principal drainage Valley

Data Element Name	AE	Values	Notes
WATERSHED			
DATE	X	60-75	Year of latest release of correlated soil survey by SCS
REGIONALBASIN		UM,GL,...	Upper Mississippi, Great Lakes,...
MAINWATERSHED		RK,SL,...	Rock River, Skokie Lagoon, ...
SUBWATERSHED		21C,0005,...	Kishwaukee River subwatershed, Chicago River (West fork, North branch, Skokie Ditch),...
TRIBUTARY		03,04,...	Rush Creek, Coon Creek,...

WELL

All water wells recorded on well logs - not necessarily all wells.

Data Element Name	AE	Values	Notes
WELL			
DATE	X	0-75	Year well was drilled (0 means before 1901)
DEPTH	X	0-2047	Surface of ground to bottom of well in feet
WATERLEVEL	X	0-999	Surface of ground to water level in feet (0 means flowing or artesian)
OUTPUT	X	0-2047	Gallons per minute
OWNERSHIP		I,M,D," "	Industrial, Municipal, Domestic
PRODUCINGUNIT		BR,DT," "	Bedrock or Glacial Drift
MEASURED		Y,N," "	Is the well periodically measured?
FREQUENCY	X	0-63	In weeks, frequency the well is measured
GEOPHYSICAL		Y,N," "	Is a geophysical study available?
STUDY		7108,41122, 622,...	Study number
PUMPTEST		Y,N," "	Has a pump test been performed?
ABANDONED		Y,N," "	Officially abandoned?
CHECKED		Y,N," "	Location verified by on-site visit?

Appendix D. The NARIS Language in BNF

BNF (Backus-Naur Form) is a notation used in the field of computer science to define the syntax of computer languages. The NARIS language, being a computer language,is presented definitively in BNF in this appendix. One may consider the NARIS User Manual as being the authority for the language without reading this appendix; any conflict, however, between the way a language construct is described in the manual and the way it is defined in this appendix is resolved in favor of its definition here.

Left and right broken brackets, "<" and ">", contain a sequence of characters representing a NARIS language construct. Each language construct is defined by a metalinguistic (metalinguistic - from metalanguage, a symbolic language used to define another language) formula. For example, the formula

<request>::= <primary request> | <secondary request>

is used to define a NARIS request. The symbol "::=" means "is defined to be" and may be read as "is". The symbol "|" means "or". Thus, from the formula above, one may read the definition of <request> as being either a <primary request> or a <secondary request>.

It should be noted that any symbol other than "::=", "|", or "?" which is not enclosed in broken brackets denotes itself. Thus, the following formuala:

```
<keyword>::=  MAP | TABULATE | CALCULATE | OUTPUT
```

means that <keyword> is one of the four words MAP, TABULATE, CALCULATE, or OUTPUT and every time that <keyword> is used on the right hand side of a "::=" symbol, one of the words - MAP, TABULATE, CALCULATE, or OUTPUT - is to be substituted for <keyword>.

Juxtaposition of marks and/or language constructs in a formula signifies juxtaposition of the sequences denoted. Thus, the formula

```
<xy>::=  K | Z | <xy> K | <xy> <digit>
```

where <digit> is

```
<digit>::=  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
```

allows many values for <xy> - some of which are

```
Z
K
Z1K7
ZKKK
ZK
```

and so on. The formula for the definition of <xy> contains a recursive rule for the formation of values of <xy>. Thus, <xy> may have the value K or Z or, given a legitimate value of <xy>, another <xy> may be formed by following it with the character K or a value of <digit>.

Special mention must be made of the use of question marks , "?", in this BNF. The question mark is used to denote that the use of the word,

language construct, or symbol preceding it is optional. For example, one of the definitions of an <entity remember request> is given in the following formula:

```
<entity remember request>::=  THE ? REGION <region name> IS ? <region expression>
```

The question marks following the words "THE" and "IS" in the formula means that "THE" and "IS" are optional. The expanded form of the formula, without using question marks, would be:

```
<entity remember request>::=  REGION <region name> <region expression> |
                              THE REGION <region name> <region expression> |
                              REGION <region name> IS <region expression> |
                              THE REGION <region name> IS <region expression>
```

Some definitions contain English phrases. For example,

```
<blank>::= an empty space.
```

"an empty space." is not a value of <blank>, but an English-like definition of <blank>. These definitions occur because of the impossibility of constructing BNF definitions for <blank>, <empty>, etc.

```
<empty>::= a null string of symbols.

<blank>::= an empty space.

<class name>::= the name of a Data Class (e.g., FORESTRY).

<data element name>::= the name of a Data Element (e.g., FORESTRY COVERTYPE).

<numeric data element name>::= see section 3.3.1.3, Numeric Data Elements,
                               in the NARIS User Manual.

<data element value>::=  the value of a Data Element.

<abbreviation name>::= a string of letters and digits which begins with
                       a letter and does not exceed 35 characters.

<function name>::= same definition as <abbreviation name>.

<region name>::=  same definition as <abbreviation name>.

"<file name>"::=  any combination of characters which does not exceed 35
                  characters and which is enclosed in quotes.

<number>::= <unsigned integer> | <decimal point> <unsigned integer> |
            <unsigned integer> <decimal point> <unsigned integer>

<integer>::= <sign> <unsigned integer>

<sign>::= <empty> | + | -

<unsigned integer>::= <digit> | <unsigned integer> <digit>

<digit>::=  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

<decimal point>::=  .
```

```
<where phrase>::=  WHERE <where clause>

<where clause>::= <where term> | <where term> OR <where clause>

<where term>::= <where factor> | <where factor> AND <where term>

<where factor>::= <condition> |  (<where clause>)

<condition>::= <data element name> <equality relational>
               <data element value> |
               <class arithmetic expression> <relational>
               <class arithmetic expression> |
               <class arithmetic expression> <relational> <number> |
               <number> <relational> <class arithmetic expression>  |
               <data element name> <one of relational>
               (<data element value list>)  |
               <class arithmetic expression> <one of relational>
               (<integer list>)

<equality relational>::= IS | IS NOT |
                         IS ?  EQUAL TO ? |
                         IS ? NOT EQUAL TO ? |
                         DOES NOT EQUAL | EQUALS | EQ | NE |
                         EQL | NEQ

<one of relational>::=  IS ? ONE OF |  IS ? NOT ONE OF

<relational>::= <equality relational> |
                 IS ? NOT <gtr lss part> THAN ?
                <relational> OR <relational> |
                IS ?  <gtr lss part> THAN ?

<data element value list>::= <data element value> |
                             <data element value>, <data element value list>

<integer list>::= <integer> | <integer>, <integer list>

<gtr lss part>::=  GREATER | GT | GTR | LESS | LT | LSS
```

It should be noted that <data element name> in <condition> can be <empty> in a <where clause>; in which case, the <data element name> which was last used in the <request> will be assumed. Thus, one may request

```
FOR MYREGION TABULATE SOIL ACRES WHERE GTR 10#
```

<class arithmetic expression>

```
<class arithmetic expression>::= <class term> | <unary operator> <class term> |
                                 <class arithmetic expression> <add operator>
                                 <class term >

<unary operator>::= <add operator>

<add operator>::=  + | -

<class term>::= <class factor> |
                <class term> <multiply operator> <class factor>

<multiply operator>::=  * |  <slash >

<slash>::=  /

<class factor>::= <class primary> | <class factor> ** <class primary>

<class primary>::= <number> | <class variable> | <class function> |
                   (<class arithmetic expression>)

<class variable>::= <numeric data element name>

<class function>::=  FUNCTION [<function definition>]
                     (<class arithmetic expression>)  |
                     <function name> (<class arithmetic expression>)
```

<tract arithmetic expression>

```
<tract arithmetic expression>::= <tract term> |
                                 <unary operator> <tract term> |
                                 <tract arithmetic expression> <add operator>
                                 <tract term>

<tract term>::= <tract factor> |
                <tract term> <multiply operator> <tract factor>

<tract factor>::= <tract primary> |  <tract factor>**<tract primary>

<tract primary>::= <number> | <tract variable> | <tract function> |
                   (<tract arithmetic expression>)

<tract variable>::= <modifier> <class arithmetic expression> |
                    <modifier> <class arithmetic expression> <where phrase>

<modifier>::=  SUM | AVE | MIN | MAX | COUNT| TOTAL | AVERAGE | MINIMUM  |
               MAXIMUM

<tract function>::=  FUNCTION [<function definition>]
                     (<tract arithmetic expression>)  |
                     <function name> (<tract arithmetic expression>)
```

```
<region expression>::= <region specification> | <region specification>
                       <Venn operation> <region specification> |
                       <region specification> <where phrase>

<Venn operation>::=  INTERSECT | OVERLAP | EXCLUDE | WITHOUT | UNION |
                     COMBINE

<region specification>::= <tract specification list> |
                          THE ? REGION ? <region name>

<tract specification list>::=  THE ? <tract specification> |
                               THE ? <tract specification>,
                               <tract specification list>

<tract specification>::= <qqsecs> <sections> <townships> | <sections> <townships> |
                         <townships>

<qqsecs>::= <qqsecs>, <qqsecs> | <qqsecs> & <qqsecs> | <qqsec unit> |
          (<qqsecs>) | <qqsecs> <blank> <qqsec unit>

<qqsec unit>::=  SEQ | SWQ | NEQ | NWQ | NH | SH | EH | WH

<sections>::= <blank> <section numbers> |
              <blank> SECTION <blank> <section number> |
              <blank> SEC <blank> <section number>

<section numbers>::= <section interval> |
                     <section interval>, <section numbers>

<section interval>::=<section number>| <section number>  - <section number>

<townships>::= <blank> T <blank> ? <township number> <blank> ?
               <township direction> <blank> ? R <blank> ? <range number>
               <blank> ? <range direction>

<township direction>::=  N | S | U | L

<range direction>::=  E | W

<township number>::= <range number>::=<trnumber> |
                     <trnumber> <blank> ?  - <blank> ? <trnumber>
```

```
<section number>::=  an <unsigned integer> value from 1 - 36.

<trnumber>::=  an <unsigned integer> value from 0 - 63.
```

```
<request>::= <primary request> |<secondary request>

<primary request>::= <keyword> <object> |
                     <on phrase> <keyword> <object> |
                     <keyword> <on phrase> <object> |
                     <for clause> <keyword> <object> |
                     <keyword> <for clause> <object> |
                     <keyword> <object> <for clause> |
                     <on phrase> <for clause> <keyword> <object> |
                     <on phrase> <keyword> <for clause> <object> |
                     <on phrase> <keyword> <object> <for clause> |
                     <for clause> <on phrase> <keyword> <object> |
                     <for clause> <keyword> <on phrase> <object> |
                     <keyword> <for clause> <on phrase> <object> |
                     <keyword> <on phrase> <for clause> <object> |
                     <keyword> <on phrase> <object> <for clause>

<on phrase>::=  ON PRINTER | ON FILE "<file name>"

<keyword>::=  TABULATE | CALCULATE | MAP | OUTPUT

<object>;:= <tabulate object> | <calculate object> | <map object> |
            <output object>
```

Note that the <keyword> preceding the <object> determines the type of the <object>.

```
<for clause> ::=  FOR <region expression>

<map object>::= <tract arithmetic expression> |
                AT l:<integer> <tract arithmetic expression> |
                <tract arithmetic expression> AT l:<integer>

<output object>::= <tract arithmetic expression> |
                   <tract arithmetic expression>, <tract arithmetic expression> |
                   <tract arithmetic expression>, <tract arithmetic expression>,
                   <tract arithmetic expression>
```

```
<tabulate request>::= <data list> <data list> <where phrase> |
                      <data list> <where phrase>, <tabulate object>

<data list>::= <class name> | <data element name> |
               <class arithmetic expression> | <data list>, <data list> |
               <empty>


<calculate object>::=<calc part> |
                     <calc part> <where phrase> |
                     <where phrase> <calc part>|
                     <calc part> <by phrase> |
                     <by phrase> <calc part> |
                     <calc part> <where phrase> <by phrase> |
                     <where phrase> <calc part> <by phrase> |
                     <where phrase> <by phrase> <calc part> |
                     <calc part> <by phrase> <where phrase> |
                     <by phrase> <calc part> <where phrase> |
                     <by phrase> <where phrase> <calc part>

<calc part>::=  THE ? <calc type> OF ? <class arithmetic expression>

<calc type>::=  TOTAL | AVERAGE | PERCENTAGE | COUNT | SUM | AVE | PERCENT |
                PCT| <calc type>, AND <calc type> | <calc type>, <calc type> |
                <calc type> AND <calc type>

<by phrase>::=  BY <data element name> | BY <class arithmetic expression>
```

<entity remember request>
REGION
ABBREVIATION
FUNCTION

```
<entity remember request>::=  THE ? REGION <region name> IS ? <region expression> |
    THE ? ABBREVIATION <abbreviation name> IS ? <abbreviation  expansion> |
    THE ? FUNCTION <function name> IS ? FUNCTION ? [? <function definition>] ?

<abbreviation expansion>::=  any combination of characters except for a # sign.

<function definition>::= <coordinate pair> <coordinate pair> |
                         <coordinate pair> <function definition>

<coordinate pair>::=  (<coordinate> <adjustment indicator>, <coordinate>)

<coordinate>::= <sign> <number>

<adjustment indicator>::= <sign>
```

<secondary request>
FORGET, SAVE, MAKE, LIST,
WHAT (IS|ARE)

```
<secondary request>::= <entity remember request> |
                       FORGET <entity list> |
                       WHAT IS THE ? <entity list> |
                       WHAT ARE THE ? <entity list> |
                       WHAT IS THE ? VERSIONS |
                       WHAT ARE THE ? VERSIONS |
                       WHAT IS THE ? FUNDS |
                       WHAT ARE THE ? FUNDS |
                       WHAT IS THE ? SPACE |
                       WHAT ARE THE ? SPACE |
                       WHAT IS THE ? <what is data part> |
                       WHAT ARE THE ? <what is data part> |
                       LIST THE ? <drctry type> ? <entity type> |
                       LIST THE ? <drctry type> ? <entity type plural> |
                       LIST THE ? <drctry type> |
                       LIST THE ? DATA ? CLASSES |
                       LIST THE ? DATA ? <class name> DATA ? ELEMENTS |
                       LIST THE ? DATA ? <data element name> DATA ? VALUES |
                       MAKE ? <drctry type> <entity list> |
                       SAVE <entity list> |
                       END

<entity list>::= <entity names with type> |
                 <entity list>, <entity names with type> |
                 <entity list> AND <entity names with type>

<entity names with type>::=  THE ? <entity type> <entity name list> |
                             THE ? <entity type plural> <entity name list>

<entity name list>::= <entity name> | <entity name>, <entity name list> |
                      <entity name> AND <entity name list>

<entity name>::= <region name> | <abbreviation name> | <function name>

<drctry type>::=  PRIVATE | SEMIPUBLIC | PUBLIC

<entity type>::=  REGION | ABBREVIATION  | FUNCTION

<entity type plural>::=  REGIONS | ABBREVIATIONS | FUNCTIONS
```

```
                                        <secondary request>
                                        LIST, MAKE, WHAT (IS/ARE),
                                        PUBLIC, SEMIPUBLIC, PRIVATE


<class data part>::=  DATA ? CLASSES | <class name> |
                      <class name> DATA ? <within class part>

<within class part>::= <element data part> |
                       <element data part>, <within class part> |
                       <element data part> AND <within class part>

<element data part>::=  ELEMENTS | <data element name> |
                        <data element name> DATA ? <within element part>

<within element part>::=  VALUES | <data element value> |
                          <data element value>, <within element part> |
                          <data element value> AND <within element part>
```