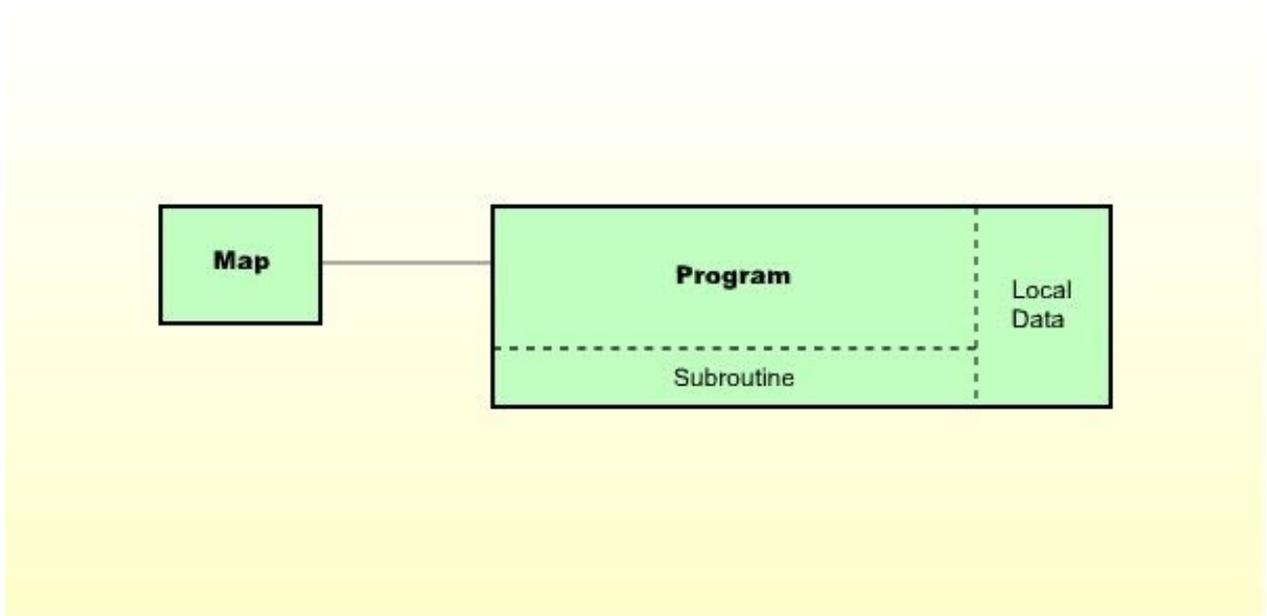


Session 1 - Creating a Program and a Map

In this session, you will use the program editor to create a Natural program. In this first session, the fields used in the program are defined as local data within the program. In addition, an inline subroutine is contained within the program.

Also, you will use the map editor to create a map used by the program.



Listed below are Steps 1 to 19 of Session 1.

Step 1

Invoke Natural according to the procedures at your site. If you get a NEXT or MORE prompt, enter the command MAINMENU. The Natural Main Menu will be displayed:

```

17:11:52          ***** NATURAL *****          2002-12-18
User SAG          - Main Menu -          Library SYSTEM

                Function

                - Development Functions
                - Development Environment Settings
                - Maintenance and Transfer Utilities
                - Debugging and Monitoring Utilities
                - Example Libraries
                - Other Products
                - Help
                - Exit Natural Session

Command ===>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
                Help          Exit                                Canc

```

Step 2

Both Natural system programs and user-written applications are stored in libraries. It may be necessary to move from one library to another in order to perform a maintenance function or work on a different application.

The field Library in the top right-hand corner shows the ID of the current library. To open another library, enter the command LOGON *library-ID* (*library-ID* being the ID of the library you want to access) in the Command line. If you have access to a library that contains copies of the sample programs used in these sessions, enter the *library-ID* of that library. By default, sample programs are provided in the system library SYSEXPG; ask your Natural administrator for details.

Step 3

On the Natural Main Menu, select Development Functions. The Development Functions menu will be displayed:

```

16:51:14          ***** NATURAL *****          2001-01-30
User SAG          - Development Functions -          Library SYSTEM
                                                Mode Reporting
                                                Work area empty

Code  Function
C    Create Object
E    Edit Object
R    Rename Object
D    Delete Object
X    Execute Program
L    List Object(s)
S    List Subroutines Used
?    Help
.    Exit

Code .. _      Type .. _
                Name .. _____

Command ==>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help           Exit                               Canc
    
```

You must be operating in structured mode to work through these sessions. If your session is in reporting mode, change the mode by entering an **S** in the first position of the Mode field. (If you cannot change the mode in this manner - it is dependent on your site - contact your Natural administrator for assistance.)

Step 4

In the course of developing application systems, Natural objects can be created and modified from the Development Functions menu.

If you wish to get help information about the functions on this menu, enter a question mark (?) in the Code field.

Below the list of functions, there are three input fields: Code, Type and Name. Further below is the Command line. You can perform a Natural function either by entering the appropriate values in the input fields, or by entering a system command in the Command line.

For example, to invoke the program editor to edit an existing program PGM01, you would enter the following in the input fields:

```

          ?    Help
          .    Exit

Code .. E      Type .. _
                Name .. PGM01_____

Command ==>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help           Exit                               Canc
    
```

The equivalent system command, entered in the Command line, would be:

```

?      Help
.      Exit

Code .. _      Type .. _
Name .. _____

Command ==> EDIT PGM01
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help      Exit                               Canc

```

If an object already exists (such as, Program PGM01 in the previous example), it is not necessary to specify its type in the Type field; this is because each object in a Natural library, regardless of its type (program, map, subroutine, etc.), must have a unique name. When you create a new object, you have to specify the type of the object in the Type field.

Once you have become familiar with the sequence of menu screens, you will be able to go directly to most of the screens you want by issuing a system command. You can enter a system command on every Natural screen that provides a Natural command prompt, such as the Command line. For further information, see the sections Natural Commands (Natural Components) and Executing Commands and Menu Functions in the Natural Fundamentals documentation.

Step 5

On the Development Functions menu, enter the code **E**. The Natural program editor will be displayed:

```

>
All      > + Program      Lib SYSTEM
.....+.....1.....+.....2.....+.....3.....+.....4.....+.....5.....+.....6.....+.....7..
0010
0020
0030
0040
0050
0060
0070
0080
0090
0100
0110
0120
0130
0140
0150
0160
0170
0180
0190
0200
.....+.....1.....+.....2.....+.....3.....+.....4.....+.....5.....+... S 0      L 1

```

- If you have access to a copy of Program PGM01, at the top of the editor screen, at the editor's command prompt (>), enter the command READ PGM01. Make sure that the program matches Program PGM01 shown on the following page.
- If you do not have access to a copy of PGM01, type in the program as illustrated in the following section. (If the

source work area is not empty, first enter the command CLEAR at the command prompt.) As you fill up a screen, type the command ADD at the command prompt for more blank lines.

When the program is complete, at the command prompt, enter the command SAVE PGM01 to store the program.

Note:

To return to the beginning of the program, at the editor's command prompt, enter the command TOP. To go to the end of the program, enter the command BOT.

Program PGM01:

```
* Example Program 'PGM01' for Natural Tutorial
* -----
DEFINE DATA
  LOCAL
  01 #NAME-START      (A20)
  01 #NAME-END        (A20)
  01 #MARK             (A1)
  01 EMPLOYEES-VIEW VIEW OF EMPLOYEES
    02 PERSONNEL-ID (A8)
    02 NAME          (A20)
    02 DEPT          (A6)
    02 LEAVE-DUE     (N2)
END-DEFINE
*
REPEAT
*
INPUT USING MAP 'MAP01'
IF #NAME-START = '.'
  ESCAPE BOTTOM
END-IF
MOVE #NAME-START TO #NAME-END
*
RD1. READ EMPLOYEES-VIEW BY NAME
  STARTING FROM #NAME-START
  THRU #NAME-END
  IF LEAVE-DUE >= 20
    PERFORM MARK-SPECIAL-EMPLOYEES
  ELSE
    RESET #MARK
  END-IF
  DISPLAY NAME 3X DEPT 3X LEAVE-DUE 3X '>=20' #MARK
END-READ
*
IF *COUNTER (RD1.) = 0
  REINPUT 'PLEASE TRY ANOTHER NAME'
END-IF
*
END-REPEAT
*
DEFINE SUBROUTINE MARK-SPECIAL-EMPLOYEES
  MOVE '*' TO #MARK
END-SUBROUTINE
*
END
```

Once you reached this point (either by writing the program yourself or by reading it into the editor), note the following aspects of this example of a Natural program:

- The first statement must always be a DEFINE DATA statement. All variables to be used in the program must be defined in this initial DEFINE DATA statement. In this example program, the variables are defined in a DEFINE DATA LOCAL statement; that is, the data definition is internal to the program.
- All statements that initiate a logical construct or processing loop (DEFINE DATA, REPEAT, IF, READ) must be ended with a corresponding END-... statement (END-DEFINE, END-REPEAT, END-IF, END-READ).
- The READ statement is marked with a statement label, namely **RD1**. Via this label it is possible to reference the statement at a later point in the program (see last IF statement).

When this program is executed, a screen appears, prompting the user to enter a name. The EMPLOYEES file is searched to locate all employees with that name. Then a report is displayed which includes the Name, Department and Leave Due of each employee with that name. Those employees who have more than 20 days leave due are marked with an asterisk (*).

The prompting screen is invoked via the INPUT USING MAP statement. The report is formatted according to information in the DISPLAY statement. The processing required to show which employees have more than 20 days leave is handled in the portion of the program starting with IF LEAVE-DUE ... Those with 20 or more days of leave due have an asterisk in the final report as a result of processing in the PERFORM statement and the DEFINE SUBROUTINE statement.

Step 6

The program contains an INPUT USING MAP statement that invokes a map named MAP01. This map is yet to be created.

At the program editor's command prompt, enter a period (.) to return to the Development Functions menu. On the Development Functions menu, in the Code field enter the function code **E** (for Edit Object) and, in the Type field, enter the type **M** (for Map):

```

      .      Exit
      Code .. E      Type .. M
      Name .. _____

Command ==>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help           Exit                                     Canc

```

The Edit Map menu will be displayed:

```

17:55:11          ***** NATURAL MAP EDITOR *****          2001-01-30
User SAG              - Edit Map -                          Library SYSTEM

          Code      Function
          ----      -
          D      Field and Variable Definitions
          E      Edit Map
          I      Initialize new Map
          H      Initialize a new Help Map
          M      Maintenance of Profiles & Devices
          S      Save Map
          T      Test Map
          W      Stow Map
          ?      Help
          .      Exit

          Code .. I      Name .. _____      Profile .. SYSPROF_

Command ==>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help      Exit Test Edit
    
```

Step 7

Enter the code **I** (Initiate a New Map) and the name MAP01. The Define Map Settings For Map screen will be displayed:

```

17:56:47          Define Map Settings for MAP          2001-01-30

Delimiters          Format          Context
-----
Cls Att CD Del      Page Size ..... 23      Device Check .... _____
T  D      BLANK    Line Size ..... 79      WRITE Statement  _
T  I      ?        Column Shift ... 0 (0/1)    INPUT Statement  X
A  D      _        Layout ..... _____
A  I      )        dynamic ..... N (Y/N)    as field default N (Y/N)
A  N      ^        Zero Print ..... N (Y/N)
M  D      &        Case Default ... UC (UC/LC)
M  I      :        Manual Skip .... N (Y/N)    Automatic Rule Rank 1
O  D      +        Decimal Char ... .      Profile Name .... SYSPROF
O  I      (        Standard Keys .. N (Y/N)
          Justification .. L (L/R)
          Print Mode ..... _

          Control Var .... _____

Apply changes only to new fields?      N (Y/N)
          Required, Partial ....
          Optional, Complete ...
          Required, Complete ...

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help      Quit                                  Let
    
```

Step 8

In the Filler Characters section of the screen, enter an underscore (_) after each of the options:

Justification .. L (L/R)	Filler Characters
Print Mode _	-----
Control Var _____	Optional, Partial _
Apply changes only to new fields? N (Y/N)	Required, Partial _
	Optional, Complete ... _
	Required, Complete ... _
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---	
Help Quit	Let

The map will use this character whenever a field has empty positions to fill within a field. (Delimiter characters can also be modified on this screen. However, for these sessions, they are left unchanged.)

Press ENTER again. The map editor screen will be displayed in split-screen format:

Ob _	Ob D CLS ATT DEL	CLS ATT DEL
.	. T D Blnk	T I ?
.	. A D _	A I)
.	. A N ^	M D &
.	. M I :	O D +
.	. O I (
.	.	
001 --010---+---+---+---030---+---+---+---050---+---+---+---070---+---		
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---		
Help Mset Exit Test Edit -- - + Full < > Let		

The top portion of the screen will not be used in this exercise. The bottom section is the editing area in which you will create the map.

During the creation of the map, you will use a number of map editor line commands and field commands.

- A line command begins with two periods (..). You enter it at the beginning of a line, and it applies to the whole line in which you enter it.
- A field command begins with one period (.). You enter it at the beginning of a field, and it applies only to the field in which you enter it.

(The sessions in Tutorial - Using the Map Editor in the Natural Editors documentation also show you how to apply these commands to more than one line/field at a time.)

Step 9

Move the cursor to the second line of the editing area and type in the following:

Please enter starting name :X(20)

When you press ENTER, the screen will look as follows:

```

Ob _                               Ob D CLS ATT DEL      CLS ATT DEL
.                                  .      T D   Blnk    T I   ?
.                                  .      A D   _      A I   )
.                                  .      A N   ^      M D   &
.                                  .      M I   :      O D   +
.                                  .      O I   (
.
001  --010---+-----+-----030---+-----+-----050---+-----+-----070---+-----

Please enter starting name :XXXXXXXXXXXXXXXXXXXXX

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help Mset Exit Test Edit -- - + Full < > Let
    
```

If the letters you typed in are automatically converted to upper case, press PF3 to return to the Edit Map menu and enter %L at the command prompt.

Step 10

Over the first three positions of the line that contains "Please enter starting name", type in the line command **..C** (Center) as shown below:

```

Ob _                               Ob D CLS ATT DEL      CLS ATT DEL
.                                  .      T D   Blnk    T I   ?
.                                  .      A D   _      A I   )
.                                  .      A N   ^      M D   &
.                                  .      M I   :      O D   +
.                                  .      O I   (
.
001  --010---+-----+-----030---+-----+-----050---+-----+-----070---+-----

..case enter starting name :XXXXXXXXXXXXXXXXXXXXX
    
```

As a result, the line will be centered:

```

Ob _          Ob D CLS ATT  DEL      CLS ATT  DEL
.            .   T  D   Blnk     T  I   ?
.            .   A  D   _         A  I   )
.            .   A  N   ^         M  D   &
.            .   M  I   :         O  D   +
.            .   O  I   (
.
001  --010---+-----+---030---+-----+---050---+-----+---070---+-----
                Please enter starting name :XXXXXXXXXXXXXXXXXXXX
    
```

Step 11

Move the cursor to the end of the new line, leave a space after the field and type in the following text:

(. to exit)

The screen will now look as follows:

```

Ob _          Ob D CLS ATT  DEL      CLS ATT  DEL
.            .   T  D   Blnk     T  I   ?
.            .   A  D   _         A  I   )
.            .   A  N   ^         M  D   &
.            .   M  I   :         O  D   +
.            .   O  I   (
.
001  --010---+-----+---030---+-----+---050---+-----+---070---+-----
                Please enter starting name :XXXXXXXXXXXXXXXXXXXX (. to exit)

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Mset  Exit  Test  Edit  --  -   +   Full  <   >   Let
    
```

Step 12

Over the first two positions of the field :XXXXXXXXXXXXXXXXXXXX, type in the field command **.E** (for extended field editing) as shown below:

```

Ob  _      Ob D CLS ATT  DEL      CLS ATT  DEL
.          .   T  D   Blnk     T   I   ?
.          .   A  D   _         A   I   )
.          .   A  N   ^         M   D   &
.          .   M  I   :         O   D   +
.          .   O  I   (
.          .
001  --010---+-----+---030---+-----+---050---+-----+---070---+-----
Please enter starting name .XXXXXXXXXXXXXXXXXXXXX (. to exit)
    
```

This will invoke the extended field editing section for the field marked with the command:

```

Fld #001                                          Fmt A20
-----
AD= MIT'_'_____  ZP= OFF      SG= OFF      HE= _____  Rls 0
AL= _____      CD=  _      CV= _____  Mod Undef
PM=  _  DF=         DY= _____
EM= _____
001  --010---+-----+---030---+-----+---050---+-----+---070---+-----
Please enter starting name .XXXXXXXXXXXXXXXXXXXXX (. to exit)
    
```

The field Fld in the upper left corner contains the field name #001. This number has been assigned automatically by Natural.

Overwrite #001 with the field name #NAME-START:

```

Fld #NAME-START                                  Fmt A20
-----
AD= MIT'_'_____  ZP= OFF      SG= OFF      HE= _____  Rls 0
AL= _____      CD=  _      CV= _____  Mod Undef
PM=  _  DF=         DY= _____
EM= _____
001  --010---+-----+---030---+-----+---050---+-----+---070---+-----
    
```

Press PF3 to leave the extended field editing section.

Step 13

The map is now complete.

Press PF4 to test the map. The map will be displayed in the form it which it will be displayed on the screen when it is invoked via the INPUT USING MAP statement in Program PGM01:

```

Please enter starting name _____ (. to exit)
    
```

Press PF3 to stop testing and return to the map editing screen.

Step 14

Press PF3 again to return to the Edit Map menu.

Enter the code **W**. The map is now stowed (that is, stored in source form and in object form) and ready to be used by the program.

To return to the Development Functions menu, enter a period (.) in the Code field.

Step 15

In the Command line of the Development Functions menu, enter the command **EDIT PGM01**.

At the top of the program editor, at the command prompt, enter the command **RUN**. This command compiles and executes Program PGM01.

The map will be displayed where the program asks you to enter a name in the input field. For demonstration purposes, enter the name **JONES**.

Based on this name, the program will produce the following output report:

Page	1				00-11-30	12:45:56
	NAME	DEPARTMENT	LEAVE	>=20		
		CODE	DUE			
	-----	-----	-----	-----		
	JONES	SALE30	25	*		
	JONES	MGMT10	34	*		
	JONES	TECH10	11			
	JONES	MGMT10	18			
	JONES	TECH10	21	*		
	JONES	SALE00	30	*		
	JONES	SALE20	14			
	JONES	COMP12	26	*		
	JONES	TECH02	25	*		

Step 16

Keep pressing **ENTER** until you get to the map again where the program asks you to enter a name. Enter a period (.) and delete the remaining characters from the input field. You will be returned to the program editor.

Step 17

Whenever you issue a **CHECK**, **RUN** or **STOW** command, the program is checked for syntax errors that would keep the program from being processed.

To introduce such an error, move the cursor to the following statement line:

```
DISPLAY NAME 3X DEPT 3X LEAVE-DUE 3X '>=20' #MARK
```

Delete the apostrophe after 20. The line now look as follows:

```
DISPLAY NAME 3X DEPT 3X LEAVE-DUE 3X '>=20 #MARK
```

At the command prompt, enter the command CHECK. The error message "Text string must begin and end on the same line." will appear:

```

>
      > + Program      PGM01      Lib SYSEXPG
      .....1.....2.....3.....4.....5.....6.....7..
0350          BY NAME
0360          STARTING FROM #NAME-START
0370          THRU #NAME-END
0380 *
0390  IF LEAVE-DUE >=20
0400          PERFORM MARK-SPECIAL-EMPLOYEEES
0410  ELSE
0420          RESET #MARK
0430  END-IF
0440 *
E 0450  DISPLAY NAME 3X DEPT 3X LEAVE-DUE 3X '>=20 #MARK
0460 *
0470  END-READ
0480 *
0490  IF *COUNTER (RD1.) = 0
0500          REINPUT 'PLEASE TRY ANOTHER NAME'
0510  END-IF
0520 *
0530 END-REPEAT
0540 *
      .....1.....2.....3.....4.....5..... S 59  L 35
NAT0305 Text string must begin and end on the same line.

```

Natural requires that a text string (in this case '>=20') must be begun and closed on the same statement line; the beginning and closing of a text string must be indicated by apostrophes. When the closing apostrophe is deleted, this condition is no longer met.

Note:

If you wish to get more information explaining the meaning of the error message, at the command prompt, you can enter a question mark (?) and the error number of the message to invoke the help system, for example, ? NAT0305.

Step 18

Type in the missing apostrophe again. At the command prompt, then enter the command CHECK again to make sure the program is now correct.

At the command prompt, then enter the command RUN. When the program has run successfully, return to the program editor again by entering a period (.).

Step 19

At the command prompt, enter the command STOW to compile the program and save both source and object form.

At the command prompt, then enter a period (.) to return to the Development Functions menu.

End of Session 1.