

Graphics Interface

This section describes how to generate Microsoft Excel graphics from Natural programs on Windows and covers the following topics:

- Generating Excel Graphics from Natural
 - CALL 'GRAPHICS'
 - CALL 'DRAW'
 - CALL 'PLOT'
-

Generating Excel Graphics from Natural

General Information

To generate Microsoft Excel graphics from Natural programs on Windows and Windows NT, you use the special programs DRAW and PLOT, which are invoked with a CALL statement. In addition the program GRAPHICS, which is also invoked with a CALL statement, establishes the communication context for DRAW and PLOT.

The parameters specified with the CALL statements are passed to Microsoft Excel. The order of the parameters is very important: when using these programs, adhere strictly to the syntax that is described in this chapter.

The individual CALL statements are described below.

Prerequisites for Generating Excel Graphics

You can only use these CALL statements to generate Excel graphics if Excel is installed on your workstation and the Add-In natgraph.xla stored in:

Program Files\Software AG\Natural\411\Natural\Samples\sysexngr\excel
is inserted in Excel.

In addition, the environment variable "natuser" must be defined to "natgraph".

To run the examples in library "sysexngr" the SAG demo database (DB 12) must be started.

Further Examples

In addition to the example shown in this chapter, further example programs are provided in library SYSEXNGR; their names begin with "NGREX".

CALL 'GRAPHICS'

```
CALL 'GRAPHICS' { 'ON' 'OFF' }
```

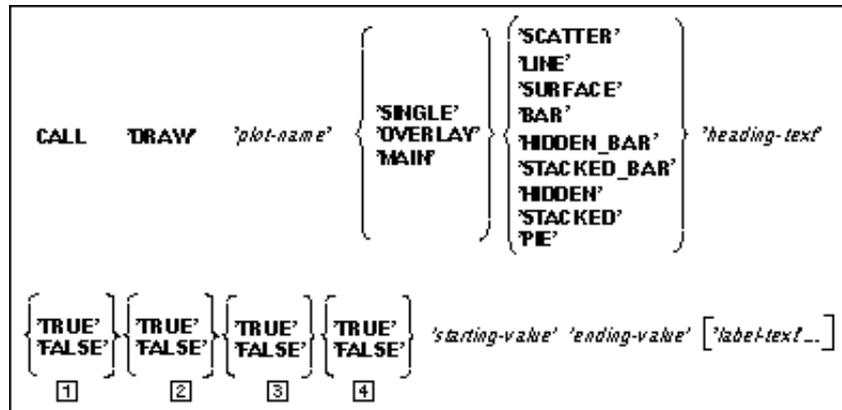
With this statement, you establish a communication context to Microsoft Excel for the statements CALL 'PLOT' and CALL 'DRAW'.

A CALL 'GRAPHICS' 'ON' statement establishes the context. This statement must be placed before the CALL 'PLOT' statement(s). The output of CALL 'PLOT' statements is available for processing by subsequent CALL 'DRAW' statements until the context is closed by a CALL 'GRAPHICS' 'OFF' statement.

Example:

```
*** Natural GRAPHICS EXAMPLE
NGREX01 *** * Plot out two sets of data DEFINE DATA LOCAL 1 EMPL-VIEW VIEW OF
EMPLOYEES 2 BIRTH 2 SALARY (1) 1 #BIRTH-START (D) 1 #BIRTH-ALL (A8) 1 REDEFINE
#BIRTH-ALL 2 #BIRTH (N8) 1 #TODAY-ALL (A8) 1 REDEFINE #TODAY-ALL 2 #TODAY (N8)
1 #AGE (N3) 1 #SALARY (N9) END-DEFINE * CALL 'GRAPHICS' 'ON' * MOVE EDITED *DATX
(EM=YYYYMMDD) TO #TODAY-ALL MOVE EDITED '31121950' TO #BIRTH-START (EM=DDMMYYYY)
* READ EMPL-VIEW BY BIRTH STARTING From #BIRTH-START MOVE EDITED BIRTH (EM=YYYYMMDD)
TO #BIRTH-ALL #AGE:= (#TODAY-#BIRTH) / 10000 #SALARY:= SALARY (1) * CALL 'PLOT'
'AGE-SALARY' 'Fixed/Salary' 'AVER' #SALARY 'Age' #AGE CALL 'PLOT' 'AGE-NUMBER'
'Number' 'COUNT' #AGE 'Age' #AGE * END-READ * * Draw out the two sets of data
using different types of chart CALL 'DRAW' 'AGE-NUMBER' 'SINGLE' 'LINE'
*** NGR-100-010 NGREX01 PLOT 1 *** FALSE FALSE TRUE TRUE 0 0 CALL 'DRAW' 'AGE-SALARY'
'SINGLE' 'SURFACE' 'Fixed/Salary by Age' FALSE FALSE TRUE TRUE 0 0 *
CALL 'GRAPHICS' 'OFF' END
```

CALL 'DRAW'



With the CALL 'DRAW' statement you define how data are to be output graphically.

A chart is drawn on the basis of the data accumulated in a previous CALL 'PLOT' statement. You reference a CALL 'PLOT' statement by specifying in the CALL 'DRAW' statement the same unique *plot-name* you specified with the respective CALL 'PLOT' statement.

If you wish the same data to be displayed in different charts, you use multiple CALL 'DRAW' statements which refer to the same CALL 'PLOT' statement. However, in one CALL 'DRAW' statement you can use the data from one CALL 'PLOT' statement only, that is, you cannot refer to more than one CALL 'PLOT' statement.

Note: If a program contains several CALL 'DRAW' statements, you can use the function "Leave Chart and Continue" from the Microsoft Excel menu "NatGraph" to "jump" from one chart to the next.

Operands of the CALL 'DRAW' Statement

The operands must be specified in exactly the order they are presented here (see also the syntax diagram above).

'plot-name'

With the *plot-name*, you refer to a specific CALL 'PLOT' statement. The *plot-name* you specify (enclosed in apostrophes) with the CALL 'DRAW' statement must be identical to the one specified in the corresponding CALL 'PLOT' statement. In this way, you identify the data which are to be used to produce the chart. You can also use a variable for *plot-name*.

Mode of Charts - 'SINGLE', 'OVERLAY', 'MAIN'

With this operand, you specify the mode in which a chart is to be drawn. The value you specify must be enclosed in apostrophes. The following values are available:

'SINGLE'	A single (separate) chart is drawn. If the program contains multiple CALL 'DRAW' statements containing a 'SINGLE' operand, the resulting charts are not overlaid, but are produced one after the other as separate charts.
'OVERLAY'	Multiple charts are combined into one chart with one overlaying the other (only applies when the program contains multiple CALL 'DRAW' statements and a previous CALL 'DRAW' statement contains the 'MAIN' operand, see below).
'MAIN'	When you have more than one CALL 'DRAW' statement and you wish to overlay the charts, use the 'MAIN' mode for the CALL 'DRAW' statement to indicate that the chart that results from this statement is to be the "main" chart, which is then overlaid with the chart produced by the CALL 'DRAW' statement with the mode 'OVERLAY' (see above).

Example of 'MAIN' and 'OVERLAY':

```
... * Draw out the two sets of data using different types
of chart CALL 'DRAW' 'SUB2' 'MAIN' 'BAR' '** NGR-100-010 Example: NGREX13
**' FALSE FALSE TRUE TRUE 0 60 CALL 'DRAW' 'SUB1' 'OVERLAY' 'LINE' ' ' FALSE FALSE
TRUE TRUE 0 0 ...
```

Types of Charts

Nine different types of charts are available with Natural.

With this operand, you specify the type of chart to be drawn. The value you specify must be enclosed in apostrophes. The following values are available:

- 'SCATTER'
- 'LINE'
- 'SURFACE'
- 'BAR'
- 'HIDDEN_BAR'
- 'HIDDEN'
- 'STACKED_BAR'
- 'STACKED'
- 'PIE'

SCATTER

In a scatter plot, the data are displayed as discrete points.

LINE

In a line graph, the data are displayed as points connected by a line.

SURFACE

In a surface chart, the data are displayed as in a line graph, and in addition the space between the x-axis and the line of coordinates is shaded.

BAR

In a bar chart, the data are displayed as vertical bars.

HIDDEN_BAR

In a hidden bar chart, the data are displayed as vertical bars; however, whereas a normal bar chart displays the bars for each x-value next to each other, in a hidden bar chart they are displayed one "hidden" behind the other. This type of chart is only applicable if you have more than one set of *y-values* for each *x-value*.

HIDDEN - no bar

In a hidden chart, the data are displayed as points joined by a line. The lines are displayed one above the other.

STACKED_BAR

In a stacked bar chart, the data are displayed as vertical bars; however, whereas a normal bar chart displays the bars for each x-value next to each other, in a stacked bar chart they are displayed one "stacked" on top of the other. This type of chart is only applicable if you have more than one set of *y-values* for each *x-value*.

STACKED - no bar

In a stacked chart, the data are displayed as points joined by a line. The lines are displayed one above the other.

PIE

In a pie chart, the data are displayed as segments of a circle.

'heading-text'

Here you can specify a heading for a chart. The *heading-text* you specify (enclosed in apostrophes) is output centered above the chart.

If you wish to have a heading which occupies multiple lines, you can use a semicolon (;) within the *heading-text* to cause a line advance: the part of the *heading-text* which follows the semicolon will then be output on the next line.

Example of Specifying a Heading that Occupies Multiple Lines:

```
.. CALL
'DRAW' 'AGE-NUMBER' 'SINGLE' 'LINE' '** NGR-100-01 NGREX01 PLOT 1 **;' - 'Example
of Line Chart' FALSE FALSE TRUE TRUE 0 0...
```

The heading can also be defined by a variable in the program.

Example of Specifying a Heading via a Variable:

```
... CALL
'DRAW' 'AGE-NUMBER' 'SINGLE' 'LINE' #HEADING FALSE FALSE TRUE TRUE 0 0
...
```

If you wish a chart to be untitled, you specify a blank enclosed in apostrophes (' ') as the *heading-text*.

Example of Specifying a Blank Heading:

```
... CALL
'DRAW' 'AGE-NUMBER' 'SINGLE' 'LINE' ' ' FALSE FALSE TRUE TRUE 0 0 ...
```

1. PERCENT Operand

This operand only applies to pie charts. For all other types of chart, set this operand to 'FALSE'.

The PERCENT operand is used for pie charts where the data to be displayed are percentage figures.

'TRUE'	PERCENT check is carried out.
'FALSE'	PERCENT check is not carried out.

When the PERCENT operand is set to "TRUE" the total for all sectors of the pie must not exceed 100; if it does, the PERCENT operand will be ignored and an appropriate error message be issued at runtime. If the total is less than 100%, an incomplete circle will be drawn.

If the PERCENT operand is set to "FALSE", Natural assumes that the data to be displayed are absolute figures. The sum of the figures is assumed to be 100%, that is, a full circle will be drawn and the size of each pie segment reflects the respective value in relation to the sum of values.

2. VALUES Operand

This operand only applies to pie charts and bar charts (normal, hidden and stacked ones). For all other types of chart, set this operand to 'FALSE'.

'TRUE'	VALUES are displayed.
'FALSE'	VALUES are not displayed.

VALUES Operand for a Bar Chart

When this operand is specified (TRUE) for a bar chart, the y-axis values are displayed above the individual bars.

VALUES Operand for a Pie Chart

When this operand is specified (TRUE) for a pie chart, figures indicating the size of each sector of the pie chart are displayed. The displayed size values always reflect the *relative sizes* of pie segments in relation to the total pie.

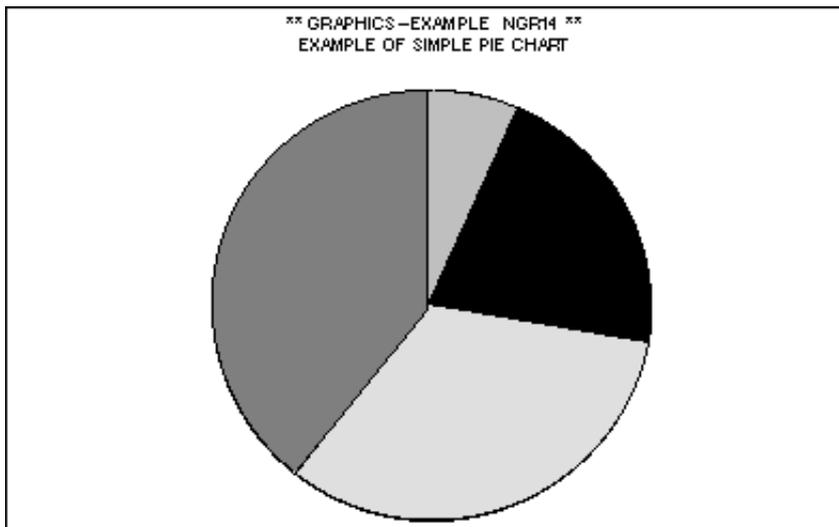
Interaction of VALUES Operand and PERCENT Operand

If the PERCENT operand is set to "FALSE", the values displayed will always add up to 100%, even if the total of all sectors is less than 100 in absolute figures. A full circle is drawn, and the values displayed do not show the actual absolute values for each segment but are in fact percentage figures indicating the relative size of each segment in relation to the total of the pie.

If the PERCENT operand is set to "TRUE", the values displayed still indicate the relative size of each segment; the values displayed are, however, identical to the actual values of the dependent variable because the actual values are (assumed to be) percentage figures.

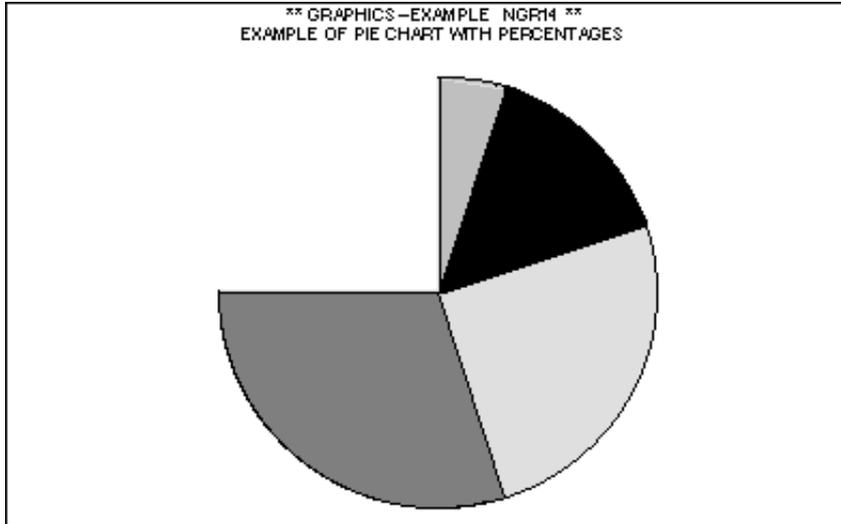
Pie Chart with PERCENT and VALUES Operands set to 'FALSE'

The values accumulated for the dependent variable are 5, 15, 25 and 30. Since these are absolute figures and not percentage figures, the PERCENT operand is not used. Natural considers the sum of the values to be 100% and therefore draws a complete circle with the size of each segment reflecting the respective value in relation to the sum of the values.



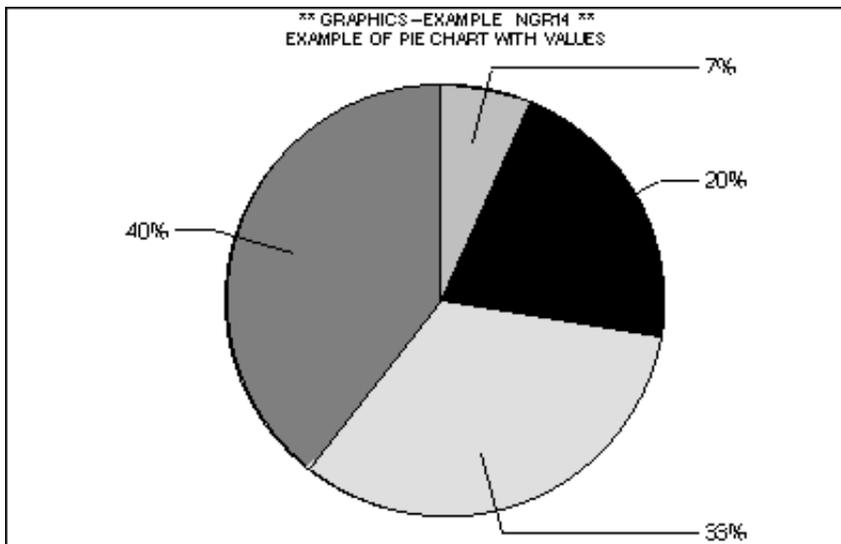
Pie Chart with PERCENT Operand set to 'TRUE'

This chart uses a PERCENT operand because the values accumulated for the dependent variable are percentage figures. As these figures do not add up to 100%, an incomplete circle is drawn.



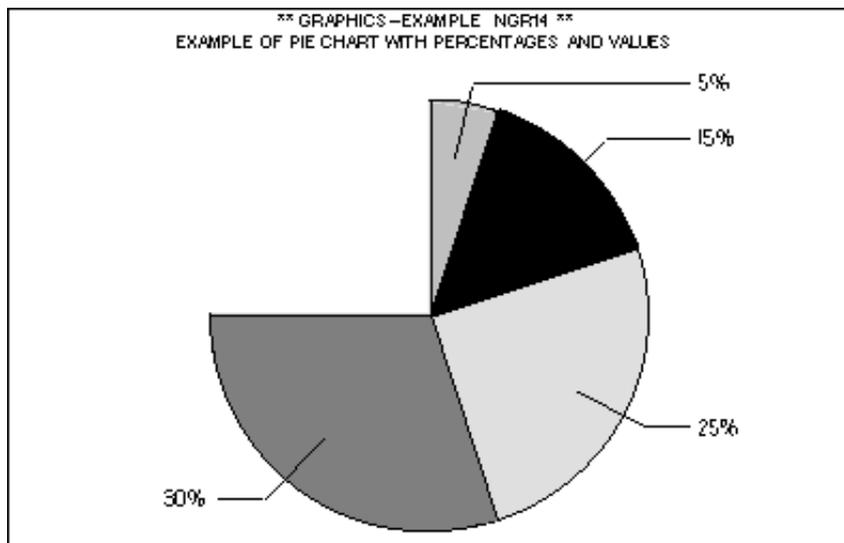
Pie Chart with VALUES Operand set to 'TRUE'

The values accumulated for the dependent variable are 5, 15, 25 and 30. Since these are absolute figures and not percentage figures, the PERCENT operand is not used. Natural considers the sum of the values to be 100% and therefore draws a complete circle with the size of each segment reflecting the respective value in relation to the sum of the values. The VALUES operand writes the segment sizes next to the segments.



Pie Chart with PERCENT and VALUES Operands set to 'TRUE'

This chart uses a PERCENT operand because the values accumulated for the dependent variable are percentage figures. The VALUES operand writes the segment sizes - which in this case correspond to the actual values accumulated, as these are percentages - next to the segments.



3. Operand for the X-Axis

'TRUE'	X-axis is displayed.
'FALSE'	X-axis is suppressed.

4. Operand for the Y-Axis

'TRUE'	Y-axis is displayed.
'FALSE'	Y-axis is suppressed.

'starting-value' & 'ending-value' - RANGE Operand

If you specify "0" for both *starting-value* and *ending-value*, the value ranges of the axes are set dynamically according to the values accumulated by the CALL 'PLOT' statement.

If you wish the y-axis to cover a specific range of values, you can specify a *starting-value* and an *ending-value* for the desired range.

'label-text'

This operand is optional. The *label-text* is used to provide labels for the scale marks of a chart's x-axis.

You specify the labels as individual character strings, each enclosed in apostrophes and separated from each other by a blank:

'label-text1' 'label-text2' 'label-text3'

To achieve proper scale mark labeling, specify a *label-text* for each x-value. If you do not wish to have a label for a scale mark, specify an empty string (' ') as the label.

Instead of specifying strings of alphanumeric text constants as *label-text*, you can also specify a single variable, and then assign the labels as values to this variable. The value of the variable will be one string of all labels to be used with the individual labels separated from each other by commas.

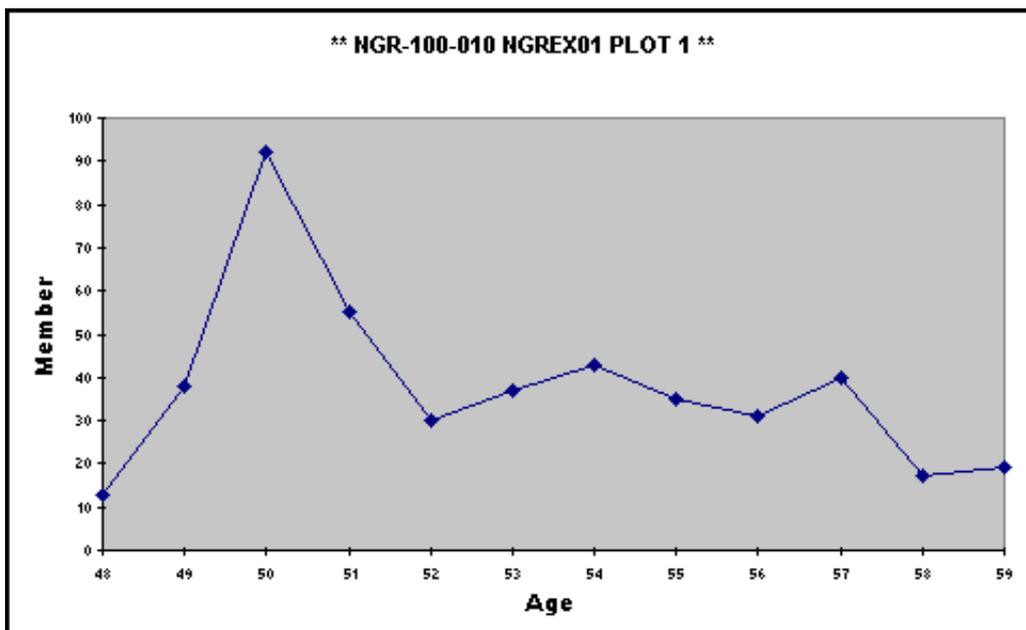
If you omit the *label-text* operand, the scale marks are labelled automatically according to the x-axis values.

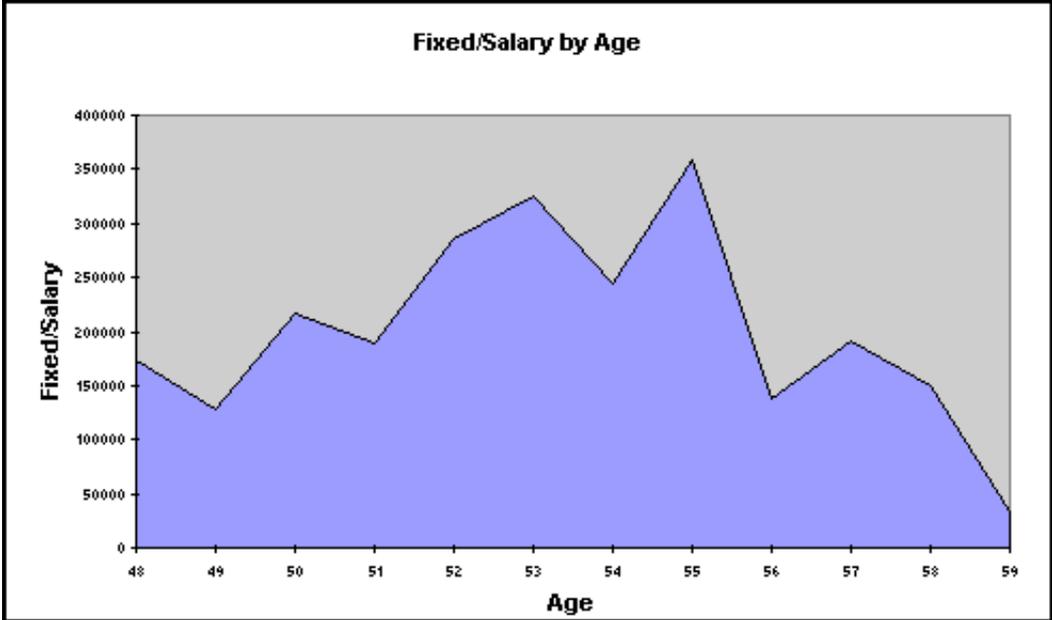
Example of CALL 'DRAW' Statements:

```

*** Natural GRAPHICS EXAMPLE
NGREX01 *** * Plot out two sets of data DEFINE DATA LOCAL 1 EMPL-VIEW VIEW OF
EMPLOYEES 2 BIRTH 2 SALARY (1) 1 #BIRTH-START (D) 1 #BIRTH-ALL (A8) 1 REDEFINE
#BIRTH-ALL 2 #BIRTH (N8) 1 #TODAY-ALL (A8) 1 REDEFINE #TODAY-ALL 2 #TODAY (N8)
1 #AGE (N3) 1 #SALARY (N9) END-DEFINE * CALL 'GRAPHICS' 'ON' * MOVE EDITED *DATX
(EM=YYYYMMDD) TO #TODAY-ALL MOVE EDITED '31121950' TO #BIRTH-START (EM=DDMMYYYY)
* READ EMPL-VIEW BY BIRTH STARTING From #BIRTH-START MOVE EDITED BIRTH (EM=YYYYMMDD)
TO #BIRTH-ALL #AGE:= (#TODAY-#BIRTH) / 10000 #SALARY:= SALARY (1) * CALL 'PLOT'
'AGE-SALARY' 'Fixed/Salary' 'AVER' #SALARY 'Age' #AGE CALL 'PLOT' 'AGE-NUMBER'
'Number' 'COUNT' #AGE 'Age' #AGE * END-READ * * Draw out the two sets of data
using different types of chart CALL 'DRAW' 'AGE-NUMBER' 'SINGLE' 'LINE'
** NGR-100-010 NGREX01 PLOT 1 ** FALSE FALSE TRUE TRUE 0 0 CALL 'DRAW' 'AGE-SALARY'
'SINGLE' 'SURFACE' 'Fixed/Salary by Age' FALSE FALSE TRUE TRUE 0 0 *
CALL 'GRAPHICS' 'OFF' END

```





CALL 'PLOT'

With the CALL 'DRAW' statement you define how data are to be output graphically. With the CALL 'PLOT' statement you specify which data are to be accumulated to produce the graph, and you also determine the point in the program at which these data are to be captured.

The data to be collected are specified by an independent variable (the *x-value*) and a dependent variable (the *y-value*).

A PLOT statement is executed at the point at which it is encountered in the program. The data point represented by the current values of the dependent and independent variables is saved so that it can be used when an associated CALL 'DRAW' statement is executed.

'plot-name'

plot-name is a user-defined name by which you identify the PLOT data in the program, and is used to refer to the accumulated data in subsequent CALL 'DRAW' statements.

plot-name must be enclosed in apostrophes, it must be unique within a program, it may be 1 to 32 characters long, and it must conform to the naming conventions for user-defined variables (as described in the Natural Reference documentation).

'y-text'

This text (which must be enclosed in apostrophes) is used as a title for the y-axis.

If you wish the y-axis to be untitled, specify a blank as *y-text* (in apostrophes): ' '.

'MAX', 'MIN', 'NMIN', 'COUNT', 'NCOUNT', 'AVER', 'NAVER', 'SUM', 'TOTAL'

The data values for the y-variable are accumulated to evaluate one of the following Natural system functions: MAX, MIN, NMIN, COUNT, NCOUNT, AVER, NAVER, SUM, TOTAL. (For details on these system functions, see the Natural Reference documentation.)

y-value

y-value operand defines the value to be used for the dependent variable.

'x-text'

This text (which must be enclosed in apostrophes) is used as a title for the x-axis.

If you wish the x-axis to be untitled, specify a blank as *x-text* (in apostrophes): ' '.

x-value

x-value defines the value to be used for the independent variable.

'ciomp-text'

This operand is optional. If you specify *ciomp-text* (enclosed in apostrophes) before the *comp-value*, this *ciomp-text* is used to generate a default chart heading if no *heading-text* is specified in the CALL 'DRAW' statement.

If you do not wish a text to be used, specify a blank as *ciomp-text* (in apostrophes): ' '.

comp-value

This operand is optional. The *comp-value* is used to specify a component by which the y-values are subdivided. This allows you to draw multiple sets of data on the same chart: for each value of the variable *comp-value* a separate set of y-values per x-value is determined, and all sets are displayed on one chart.

To uniquely identify each component, a legend is automatically generated using the values of the component variable and is displayed to the right of the graph.

Example of CALL 'PLOT' Statement (with comp-value):

```

*** Natural GRAPHICS EXAMPLE NGREX03 *** * First gather
the plot data DEFINE DATA LOCAL 1 EMPL-VIEW VIEW OF EMPLOYEES 2 BIRTH 2 SALARY
8(1) 2 SEX 1 #TODAY-ALL (A8) 1 REDEFINE #TODAY-ALL 2 #TODAY (N8) 1 #BIRTH-ALL
(A8) 1 REDEFINE #BIRTH-ALL 2 #BIRTH 1 #BIRTH-FROM (D) 1 #BIRTH-TO (D) 1 #AGE (N3)
1 #SALARY (N9) END-DEFINE * CALL 'GRAPHICS' 'ON' * MOVE EDITED *DATX (EM=YYYYMMDD)
TO #TODAY-ALL MOVE EDITED '16021953' TO #BIRTH-FROM (EM=DDMMYYYY) MOVE EDITED
'16021959' TO 'BIRTH-TO (EM=DDMMYYYY) * FIND EMPL-VIEW WITH BIRTH = #BIRTH-FROM
THRU #BIRTH-TO SORTED BY BIRTH DESC MOVE EDITED BIRTH (EM=YYYYMMDD) TO #BIRTH-ALL
#AGE:=(#TODAY-#BIRTH) / 10000 #SALARY:=SALARY (1) CALL 'PLOT' 'SALARY-SEX-AGE'
'Average Salary' 'AVER' #SALARY 'Age' #AGE ' ' SEX END-FIND * * Now draw
it out correctly labeling each bar CALL 'DRAW' 'SALARY-SEX-AGE' 'SINGLE' 'BAR'
'*** NGR-100-010 Example: NGREX03 ***' FALSE FALSE TRUE TRUE 0 0 * CALL 'GRAPHICS'
'OFF' * END
    
```

In this example, the *comp-value* is SEX: the chart shows the number of persons of a certain age separately for each value of the field SEX (M = male, F = female).

