

User-Defined Variables

User-defined variables are fields which you define yourself in a program. They are used to store values or intermediate results obtained at some point in program processing for additional processing or display.

The following topics are covered:

- Defining User-Defined Variables
 - Names of User-Defined Variables
 - Format and Length of User-Defined Variables
-

Defining User-Defined Variables

You define a user-defined variable by specifying its name and its format/length in the DEFINE DATA statement.

Example:

In this example, a user-defined variable of alphanumeric format and a length of 10 positions is defined with the name #FIELD1.

```
DEFINE DATA LOCAL
1 #FIELD1 (A10)
. . .
END-DEFINE
```

Names of User-Defined Variables

When working with user-defined variables, the following naming conventions must be met.

Length of Variable Names

The name of a user-defined variable may be 1 to 32 characters long.

You can use variable names of over 32 characters (for example, in complex applications where longer meaningful variable names enhance the readability of programs); however, only the first 32 characters are significant and must therefore be unique, the remaining characters will be ignored by Natural.

Limitations of Variable Names

The name of a user-defined variable must not be a Natural reserved word.

Within one Natural program, you must not use the same name for a user-defined variable and a database field, because this might lead to referencing errors (see Qualifying Data Structures).

Characters Allowed in Variable Names

The name of a user-defined variable can consist of the following characters:

Character	Explanation
A - Z	alphabetical characters (upper and lower case)
0 - 9	numeric characters
-	hyphen
@	at sign
_	underline
/	slash
\$	dollar sign
§	paragraph sign
&	ampersand
#	hash/number sign
+	plus sign (only allowed as first character)

First Character of Variable Names

The first character of the name must be one of the following:

- an upper-case alphabetical character
- #
- +
- &

If the first character is a "#", "+" or "&", the name must consist of at least one additional character.

Variables in a GDA with a "+" as first character must be defined on Level 01. Other levels are only used in a redefinition.

"+" as the first character of a name is only allowed for application-independent variables (AIVs) and variables in a global data area. Names of AIVs must begin with a "+".

"&" as the first character of a name is used in conjunction with dynamic source program modification (see the RUN statement in the Natural Statements documentation), and as a dynamically replaceable character when defining processing rules (see the map editor description in your Natural Editors documentation).

Special Considerations Regarding the Case of Characters in Variable Names

On Windows and UNIX, lower-case characters entered as part of a variable name are internally converted to upper case. The same happens on mainframe computers if the LOWSRCE option of the COMPOPT system command is set to ON.

Lower-case characters can only be entered as the second and subsequent characters of a variable name.

On mainframe computers, lower-case characters are not translated to upper case and are therefore interpreted as being different from the respective upper-case characters, if

- the LOWSRCE option of the COMPOPT system command is set to OFF (the default value) and
- input in the editor is not translated to upper case (translation to upper case in the editor is controlled by editor profile options and by options depending on the operating system).

For example, this will cause the names #FIELD and #field to be interpreted as two different field names.

Note:

For compatibility reasons, you should not use this feature if you plan to port applications developed on mainframe computers to Windows or UNIX.

If you use lower-case characters as part of the variable name, it is highly recommended that variable names are unique regardless of their case.

Format and Length of User-Defined Variables

Format and length of a user-defined variable are specified in parentheses after the variable name.

A user-defined variable can have one of the following formats and corresponding lengths:

Format	Definable Length	Internal Length (in Bytes)
A Alphanumeric	1 - 1073741824 (1GB)	1 - 1073741824
B Binary	1 - 1073741824 (1GB)	1 - 1073741824
C Attribute Control	-	2
D Date	-	4
F Floating Point	4 or 8	4 or 8
I Integer	1, 2 or 4	1, 2 or 4
L Logical	-	1
N Numeric (unpacked)	1 - 29	1 - 29
P Packed numeric	1 - 29	1 - 15
T Time	-	7

Further information is provided in User-defined Variables in the Natural Statements documentation.

Examples of User-Defined Variables

```

DEFINE DATA LOCAL
    /* 7 positions before and 2 after decimal point.
    /* and 1 sign position.
...
END-DEFINE

```

Note:

When a user-defined variable of format P is output with a DISPLAY, WRITE, or INPUT statement, Natural internally converts the format to N for the output.