

Rules and Naming Conventions

This section describes Natural-specific rules and naming conventions.

- Standard Character Settings
 - Object Naming Conventions
 - Library Naming Conventions
 - Naming Conventions for User-Defined Variables
-

Standard Character Settings

The tables in this section list the standard character settings that apply to Natural. These settings comply with the IBM code page 1140 (US English), which determines the assignment of code points to graphic characters. Code points are represented by hexadecimal values. For further details on IBM code pages, refer to the corresponding IBM documentation.

The display control unit or emulator determines what character to display for a hexadecimal value. The use of a code page other than 1140 can result in the following:

- A character provides a hexadecimal value that is not accepted by Natural.
- A character that differs from the standard character setting provides the same hexadecimal value defined for the standard character.

Examples:

The character @ (commercial at) used with the German code page 1141 does not provide the hexadecimal value 7C.

The character £ (pound sign) used with the UK English code page 1146 provides the same hexadecimal value as the \$ (dollar sign) character used with the US English code page 1140.

The character § (section sign) used with the German code page 1141 provides the same hexadecimal value as the character @ used with the US English code page 1140.

Object Naming Conventions

This section describes the naming conventions that apply when saving and/or cataloging a Natural object in a Natural system file.

The name of a Natural object can be 1 to 8 characters (listed in the following table) where the *first* character must be one of the following:

- an upper-case alphabetical character

- a number sign (#)
- a plus sign (+)

If the first character is a number sign (#) or a plus sign (+), the name must consist of at least one additional character.

Exception:

The name of a Natural DDM can be 1 to 32 characters (listed in the following table) where the *first* character must be an upper-case alphabetical character.

The name of a Natural object can consist of the following characters:

Character	EBCDIC Hex Value	ISO Character Name	Remark
A - Z	C1-C9 D1-D9 E2-E9	Latin capital letter A - Z	Upper-case alphabetical character
0 - 9	F0-F9	Digit zero - digit nine	Numeric character
-	60	Hyphen-minus	Hyphen
_	6D	Low line	Underscore
/	61	Solidus	Slash
@	7C	Commercial at	
\$	5B	Dollar sign	
&	50	Ampersand	Only allowed in language codes <i>See also Defining the Language of a Natural Object in the Programming Guide.</i>
#	7B	Number sign	Hash sign
+	4E	Plus sign	Only allowed as the first character

Library Naming Conventions

This section describes the naming conventions that apply to a Natural library.

The name (ID) of a library can be 1 to 8 characters and must *not* start with "SYS". The prefix "SYS" is reserved for Natural system libraries.

A library name must start with an upper-case alphabetical character. Any other alphabetical character must also be upper case. A library name must *not* contain any blank characters.

A library name can consist of the following characters:

Character	EBCDIC Hex Value	ISO Character Name	Remark
A - Z	C1-C9 D1-D9 E2-E9	Latin capital letter A - Z	Upper-case alphabetical character
0 - 9	F0-F9	Digit zero - digit nine	Numeric character
-	60	Hyphen-minus	Hyphen
_	6D	Low line	Underscore Not allowed as the last character

Naming Conventions for User-Defined Variables

This section describes the naming conventions that apply to a user-defined variable:

- Length of Variable Names
- Limitations of Variable Names
- Characters Allowed in Variable Names
- First Character of Variable Names
- Case of Characters in Variable Names

For further information on user-defined variables, refer to the section *User-Defined Variables* in the *Programming Guide*.

Length of Variable Names

The name of a user-defined variable can 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 keyword.

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* in the *Programming Guide*).

Characters Allowed in Variable Names

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

Character	EBCDIC Hex Value	ISO Character Name	Remark
A - Z	C1-C9 D1-D9 E2-E9 81-89 91-99 A2-A9	Latin capital and/or small letter A - Z	Upper-case and/or lower-case alphabetical character Lower-case <i>not</i> allowed as the first character
0 - 9	F0-F9	Digit zero - digit nine	Numeric character
-	60	Hyphen-minus	Hyphen
_	6D	Low line	Underscore
/	61	Solidus	Slash
@	7C	Commercial at	
\$	5B	Dollar sign	
&	50	Ampersand	
#	7B	Number sign	Hash sign
+	4E	Plus sign	Only allowed as the first character

First Character of Variable Names

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

Character	EBCDIC Hex Value	ISO Character Name	Remark
A - Z	C1-C9 D1-D9 E2-E9	Latin capital letter A - Z	Upper-case alphabetical character
&	50	Ampersand	
#	7B	Number sign	Hash sign
+	4E	Plus sign	

If the first character is a number sign (#), a plus sign (+) or an ampersand (&), the name must consist of at least one additional character.

Variables in a global data area (GDA) with a plus sign (+) as the first character must be defined at Level 1; see also *Global Data Area* in the *Programming Guide*. Other levels are only used in a redefinition.

A plus sign (+) as the first character of a name is only allowed for application-independent variables (AIVs) and variables in a global data area (GDA).

Names of AIVs must begin with a plus sign (+); see also *Defining Application-Independent Variables* in the description of the DEFINE DATA statement in the *Statements* documentation.

An ampersand (&) as the first character of a name is used in conjunction with dynamic source program modification (see the *RUN* statement in the *Statements* documentation), and as a dynamically replaceable character when defining processing rules; see the relevant description in the *Map Editor* documentation.

Case of Characters in Variable Names

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

Lower-case characters entered as part of a variable name are internally converted to upper case if the *LOWSRCE* option of the *COMPOPT* system command (see also the *System Commands* documentation) is set to *ON*.

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 Natural applications developed on mainframe computers to Windows, UNIX or OpenVMS. 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.