

# Natural System Error Messages 0401-0450

## NAT0401: Invalid program name.

Text	Invalid program name.
Expl.	A program name must begin with an alphabetical character and must not be longer than 8 characters.
Actn.	Use correct program name.

## NAT0402: "text" for EDIT command does not end with apostrophe.

Text	"text" for EDIT command does not end with apostrophe.
Expl.	. "text" can be used in the line editor to position to a specific line beginning with that text string. .S"text" can be used to scan the source program for the specified text and select all lines containing the text.
Actn.	Check command and correct error.

## NAT0403: Invalid line editor subcommand.

Text	Invalid line editor subcommand.
Expl.	The following subcommands may be used in the line editor: .B position to bottom .Cn(m) copy m lines starting from line identified with n .C"text"(m) copy the line beginning with "text" to position m .D delete line .D(n) delete n lines .E terminate line .I insert line .I(program) insert program .Mn move lines identified by n .M"text"(n) move n lines starting from line beginning with "text" .R(n) renumber with increment n -- default n=10 .S"text" scan for text See Natural documentation, EDT command, for use of PF keys.
Actn.	See Natural documentation, EDT command, for use of PF keys. Issue a valid command.

## NAT0404: Number for "+/-" in EDIT command invalid.

Text	Number for "+/-" in EDIT command invalid.
Expl.	" +/-n" can be used to position forward/backward in the line editor. "n" must be a numeric constant, and must immediately follow "+/-".
Actn.	Enter a valid command.

**NAT0406: Explicit index range not allowed with SORT.**

Text	Explicit index range not allowed with SORT.
Expl.	Only complete arrays can be specified in the USING clause of the SORT statement. This can be achieved by specifying "*" as index or by explicitly specifying the complete range of constants. Example: 01 ARRAY (A5/1:20) ... SORT BY ... USING ARRAY(*) or SORT BY ... USING ARRAY(1:20)
Actn.	Check program and correct error.

**NAT0407: Invalid tab notation "nT".**

Text	Invalid tab notation "nT".
Expl.	The "nT" positioning must be within the useable line size. "n" must not be zero or a value greater than the currently specified line size.
Actn.	Check program and correct error.

**NAT0408: Tab notation "nT" duplicated or invalid in VERT mode.**

Text	Tab notation "nT" duplicated or invalid in VERT mode.
Expl.	The "nT" positioning can be used to position to a specific output position. In VERT mode, positioning can only be done once between two columns. It cannot be specified for individual fields in VERT mode, because the column position can only be defined once before the VERT entry. If indentation is required, use the "nX" notation within a column.
Actn.	Check program and correct error.

**NAT0409: Position ref. to field only allowed once between columns.**

Text	Position reference to field only allowed once between columns.
Expl.	The position reference of the form "T*NAME" can only be used to position to a new column in a DISPLAY statement. Within a column in VERT mode, it is not possible to position individual fields with the "T*NAME" notation. If indentation is required within a column, use the "nX" notation.
Actn.	Check program and correct error.

**NAT0411: Print position reference must be in first DISPLAY.**

Text	Print position reference must be in first DISPLAY.
Expl.	Only the first DISPLAY statement creates a print position reference. Reference to print positions (columns) in the form "T*NAME" can only be made to variables that have been defined as columns in the first DISPLAY statement defined in the program. The short-form notation (2 characters) for database fields is not permitted.
Actn.	Correct error.

**NAT0412: "nX" or "nT" notation positions beyond line size.**

Text	"nX" or "nT" notation positions beyond line size.
Expl.	The notation "nX" or "nT" resulted in an output position that lies beyond the currently specified line size.
Actn.	Either correct the nX/nT notation or increase line size parameter.

**NAT0413: Module in source library does not exist for .I(name).**

Text	Module in source library does not exist for .I(name).
Expl.	The source module to be copied from the source library cannot be located in the current library, or the entry ".I(name)" was not specified correctly.
Actn.	Re-issue command using correct source module name.

**NAT0415: "T\*" and "P\*" notation not permitted in PRINT statement.**

Text	"T*" and "P*" notation not permitted in PRINT statement.
Actn.	Do not use "T*" or "P*" in a PRINT statement.

**NAT0416: "x/y" notation not permitted with PRINT statement.**

Text	"x/y" notation not permitted with PRINT statement.
Actn.	Do not use the notation "x/y" in a PRINT statement.

**NAT0417: Variable index range not permitted with READ WORK FILE.**

Text	Variable index range not permitted with READ WORK FILE.
Actn.	Check program and correct error.

**NAT0418: "RETURN" is not permitted in structured mode.**

Text	"RETURN" is not permitted in structured mode.
Expl.	The RETURN statement is permitted in reporting mode only. In structured mode, the end of a subroutine definition must be indicated with the statement END-SUBROUTINE.
Actn.	Check program and correct error.

**NAT0419: "USING" is required in structured mode.**

Text	"USING" is required in structured mode.
Expl.	When the SORT statement is used in structured mode, the USING clause has to be used to indicate the fields which are to be written to intermediate storage. If USING KEYS is specified, all sort-key fields are written to intermediate storage. USING operand2 ... may be used to indicate fields in addition to the sort-key fields which are to be written to intermediate storage. By default in structured mode, the fields specified in the BY clause are NOT written to intermediate sort storage.
Actn.	Check program and correct error.

**NAT0422: Invalid or misplaced statement in external subroutine.**

Text	Invalid or misplaced statement in external subroutine.
Expl.	An external subroutine must not contain more than one DEFINE SUBROUTINE statement. The DEFINE SUBROUTINE statement block must be closed with END-SUBROUTINE (or RETURN, in reporting mode). The only statements allowed outside the DEFINE SUBROUTINE block are DEFINE DATA LOCAL/GLOBAL, which must precede the DEFINE SUBROUTINE statement, and END at the end of the subroutine.
Actn.	Correct error in external subroutine.

**NAT0423: Invalid GIVING LENGTH clause in READ WORK FILE statement.**

Text	Invalid GIVING LENGTH clause in READ WORK FILE statement.
Expl.	Possible reasons for this error are: 1. The keyword "LENGTH" is missing after "GIVING". 2. The operand is either missing or defined with an incorrect format/length (format/length must be I4).
Actn.	Check program and correct error.

**NAT0424: Invalid operand for PC "COMMAND" option.**

Text	Invalid operand for PC "COMMAND" option.
Expl.	The operand for the COMMAND option of the WRITE WORK FILE/WRITE PC/DOWNLOAD statement must be of alphanumeric format and its length must not exceed 80 characters.
Actn.	Check program and correct error.

**NAT0425: Invalid use of a qualified variable name.**

Text	Invalid use of a qualified variable name.
Expl.	Qualified variable names may only be used when referencing data variables. They cannot be used when defining variables in a DEFINE DATA statement.
Actn.	Check program and correct error.

**NAT0426: Invalid use of variable name in data structure.**

Text	Invalid use of variable name in data structure.
Expl.	In the DEFINE DATA statement, a variable name was used in defining a data structure that currently exists. The earlier variable definition was not part of another data structure.
Actn.	Check program and correct error.

**NAT0428: Invalid variable name definition in a data structure.**

Text	Invalid variable name definition in a data structure.
Expl.	<p>A variable name used in the DEFINE DATA statement already exists. The earlier definition of this variable name describes a data element that is contained within the current data structure. Non-unique variable names in a DEFINE DATA statement are referenced by qualifying the variable with the level 1 structure name. Therefore, two data structure variables with the same variable name can not co-exist if they share the same level 1 structure name. E.g.:</p> <pre> 1 GROUP 2 STRUCTURE-1 3 SAME-NAME (A1) 2 STRUCTURE-2 3 SAME-NAME (N7) &lt;== cannot be uniquely referenced. 1 OTHER-STRUCTURE 2 SAME-NAME (P4) &lt;== unique reference as "OTHER-STRUCTURE.SAME-NAME"</pre>
Actn.	<pre>2 SAME-NAME (P4) &lt;== unique reference as "OTHER-STRUCTURE.SAME-NAME"</pre> <p>Check program and correct error.</p>

**NAT0429: Index for database array incorrectly specified.**

Text	Index for database array incorrectly specified.
Expl.	When defining an array, the index specification for each dimension must not include a non self-defining range. A range which is not self-defining includes any of the following index specifications: #FIELD(#I + 1) or #FIELD(#J - 1) #FIELD(*) #FIELD(#I + #J)
Actn.	Check program and correct error.

**NAT0431: Qualifiers must be structure names on level 1.**

Text	Qualifiers must be structure names on level 1.
Expl.	To identify a variable which is defined inside a data structure, you may prefix the variable name with the name of the level-1 structure field, separated with a period. Only level-1 structure field names can be used as field qualifiers; fields defined on level 2 or higher are not permitted.
Actn.	Supply a valid qualifier field.

**NAT0434: Qualification does not uniquely identify variable.**

Text	Qualification does not uniquely identify variable.
Expl.	The qualified name list does not uniquely describe the current variable name.
Actn.	Check program and correct error.

**NAT0435: Index for database array incorrectly specified.**

Text	Index for database array incorrectly specified.
Expl.	The Natural Version 1.2 notation of defining an index range with the "-" character cannot be used to reference a database array which was defined using the Version 2.1 notation, that is the character ":". The converse is not permitted either. Neither is it permitted to mix the two notations when defining a database array. Neither are the index notations "I + 1" or "I - 2" etc. permitted when referencing a database array that was defined with the Version 1.2 notation.
Actn.	Check program and correct error.

**NAT0436: The internal size of a data structure is too big.**

Text	The internal size of a data structure is too big.
Expl.	The size of a data structure in the DEFINE DATA statement is too big.
Actn.	Reduce number of variables or number of occurrences for a variable.

**NAT0437: Constant definition missing for current structure.**

Text	Constant definition missing for current structure.
Expl.	If a variable is defined as a named constant via the keyword "CONST" in a DEFINE DATA statement, each elementary field of that structure must be defined as a constant.
Actn.	Check program and correct error.

**NAT0438: Record length must not exceed 1073741824 bytes.**

Text	Record length must not exceed 1073741824 bytes.
Expl.	The length of a record read/written with a READ/WRITE WORK FILE statement must not exceed 1073741824 bytes (= 1 GB).
Actn.	Check program and correct error.

**NAT0439: Variable-length field only allowed in WRITE WORK VARIABLE.**

Text	Variable-length field only allowed in WRITE WORK VARIABLE.
Expl.	Fields with variable length cannot be used in a WRITE WORK FILE statement without VARIABLE clause. Incorrect example: WRITE WORK FILE 1 array(1:j) Correct example : WRITE WORK FILE 1 VARIABLE array(1:j)
Actn.	Check program and correct error.

**NAT0440: DISPLAY and NEWPAGE not allowed in AT TOP/END OF PAGE.**

Text	DISPLAY and NEWPAGE not allowed in AT TOP/END OF PAGE.
Expl.	A DISPLAY or a NEWPAGE statement must not be placed within an AT TOP OF PAGE or AT END OF PAGE statement block.
Actn.	Correct error in program.

**NAT0441: RETURN or END-SUBROUTINE not allowed within a condition.**

Text	RETURN or END-SUBROUTINE not allowed within a condition.
Expl.	A RETURN or END-SUBROUTINE statement must not be placed within any of the following statement blocks: - AT START/END OF DATA - AT TOP/END OF PAGE - AT BREAK - BEFORE BREAK PROCESSING - AT END OF FILE - IF NO RECORDS FOUND - ON ERROR
Actn.	Correct error in program/subroutine.

**NAT0442: RETRY statement is not allowed within condition.**

Text	RETRY statement is not allowed within condition.
Expl.	A RETRY statement must not be placed within any of the following statement blocks: - AT START/END OF DATA - AT TOP/END OF PAGE - AT BREAK - BEFORE BREAK PROCESSING - AT END OF FILE - IF NO RECORDS FOUND - DEFINE SUBROUTINE
Actn.	Correct error in program.

**NAT0443: C routine name not found in jump table.**

Text	C routine name not found in jump table.
Expl.	CALL <MODULE> <ROUTINE> <ARGUMENTS> The routine declared by <routine> is not found by the jump table look-up routine declared in the module header csect <module>. Note that the jump table look-up routine is also written in C.
Actn.	This is an internal Natural Expert error. Please contact Software AG and describe how the error was produced.

**NAT0444: LFILE directive for Entire system file is not type Entire.**



Text	LFILE directive for Entire system file is not type Entire.
Expl.	<p>Check the LFILE directives for the Entire system files (249,250).  Check the Entire NTDB macro entry for validity.  In the NATPARAM module, the NTDB macro determines the database type:</p> <ul style="list-style-type: none"> <li>- NTDB ENTIRE,xyz (where "xyz" is the database ID)</li> </ul> <p>The NTLFILE macro in the NATPARAM module describes the Entire cross-reference and data files:</p> <ul style="list-style-type: none"> <li>- NTLFILE 249,xyz,10 (SYS1)</li> <li>- NTLFILE 250,xyz,11 (SYS2)</li> </ul> <p>Alternatively, LFILE can be specified as dynamic parameter; for example:</p> <ul style="list-style-type: none"> <li>- LFILE=(249,xyz,10),LFILE=(250,xyz,11)</li> </ul> <p>In both cases, the database handler must be declared as being of type ENTIRE. See the Natural documentation for further information.</p>
Actn.	Contact your Entire DB administrator.

**NAT0445: C buffer being used by another application.**

Text	C buffer being used by another application.
Expl.	<p>Two applications are using the same Natural buffer.  This error is detected on starting the second application and as it is desirable to protect data, the second application is not allowed to start.</p>
Actn.	The problem could be resolved if the two applications were to use separate buffers - please contact Software AG.

**NAT0446: Statement name found in attribute expression.**

Text	Statement name found in attribute expression.
Expl.	This error occurs if you have defined an attribute to a category which has the same name as a Natural keyword.
Actn.	Rename the attribute.

**NAT0447: FIND UNIQUE not allowed for Entire files.**

Text	FIND UNIQUE not allowed for Entire files.
Expl.	This statement is not supported by Entire.
Actn.	Correct error in program.

**NAT0448: C interface - get memory buffer failed.**

Text	C interface - get memory buffer failed.
Expl.	When a function written in C performs a MALLOC operation to get a work buffer, this is supplied to it by the GETMAIN# entry in the environment interface NICMAIN, which in turn gets it from the second buffer declared in the C module header. This buffer has not been defined in the Natural environment. With Natural Expert this means that no EXRSIZE or EXCSIZE could be allocated.
Actn.	Check whether there is enough space for the EXRSIZE or EXCSIZE as defined in the NATPARM module or increase the buffer size(s).

**NAT0449: C work buffer too small.**

Text	C work buffer too small.
Expl.	CALL <module> <routine> <arguments> On initialization of the C work buffer, the minimum size required is <size of extern/static> + 4K.
Actn.	This is an internal Natural Expert error, please contact Software AG.

**NAT0450: Not enough memory to run C subroutines.**

Text	Not enough memory to run C subroutines.
Actn.	This is an internal Natural Expert error, please contact Software AG.