Error Messages

Error Messages

Version 18-Nov-2020

General Error Messages

syntax error
The command in the command line contains a syntax error.

function not implemented
This command is not supported in the installed version of the debugger software. Please check www.lauterbach.com for availability of updates or contact technical support (support@lauterbach.com).

line too long
The current command line is too long to be parsed. Make the command line shorter.

too many keywords in line
The current command line has too many keywords to be parsed. Make the command line shorter.

keyword too long
The current command contains a keyword that is too long. The keyword/command is invalid.

unknown command
The command is unknown in the current context. This does not imply that the command could not be used in a different context. If working with multiple debug interfaces, be sure to select the desired device using its prompt.

command locked
The command exists but cannot be executed in the current system-state. When the system is brought to the correct state, the command will be enabled.

NOTE: when a command is locked, this does not automatically imply that subcommands were also locked (e.g. RTS.ISTAT may be locked while RTS.ISTAT.LIST is available).

subcommand expected
The command is incomplete. It requires another keyword after the last dot.

unknown keyword
The used keyword (usually an argument of the current command) is written wrong or is locked.

unknown keyword '<*name*>'
The used keyword (usually an argument of the current command or PRACTICE function) is written wrong or is locked.

keyword locked

The keyword cannot be used in the current system-state. When the system is brought to the correct state, the keyword will be enabled.

not supported or not existing option
The option is unknown or not supported in the current context.
This does not imply that the option could not be used in a different context.
If working with multiple debug interfaces, be sure to select the desired device using its prompt.
e.g. B::FLASH.TARGET 0x0++1 0x10++1 abc.bin /DualPort // isn't available in some CPU architectures.

option locked
The option exists but cannot be executed in the current system-state. When the system is brought to the correct state, the command will be enabled.

no more arguments expected
The command line contains more arguments than the current command can accept.

more arguments expected
The command line contains fewer arguments than the current command needs. The amount of arguments can depend on the current system state or also on other commands.

REPEAT command cannot directly be followed by this command. Use brackets.

missing 'code' in ON KEY 'code' 'action'.

missing 'name' in ON CMD 'name' 'action'.
Type a user-defined command name. The length of user-defined command names is limited to 9 characters. Permissible characters for user-defined command names are [0..9], [@..Z], [a..z], '_', '+' and '-'.

invalid character in 'name' of ON CMD 'name' 'action' detected.
Permissible characters for user-defined command names are [0..9], [@..Z], [a..z], '_', '+' and '-'.

missing 'command' in ON CMD 'name' EXECute 'command'

missing 'time' in ON TIME 'time' 'action'.

missing 'action' in ON TIME 'time' 'action'.

missing 'label' or 'line' for given action.

too many items
The argument list of the current command contains too many items. Try again with fewer items.

illegal combination of options
Two or more options of the current command can not be used at the same time.

value too small
The value of the entered argument is smaller than the allowed minimum value.

value too large

The value of the entered argument is larger than the allowed maximum value.

value not allowed
The value of the entered argument does not match to the allowed value ranges.

value too large - truncated to maximum value
The value of the entered argument was too large. The maximum value will be set. E.g. "`Analyzer.SIZE 805306368.`" is invalid for an ETM preprocessor with a POWER TRACE with 1GB trace memory and will be therefore truncated to "`Analyzer.SIZE 268435456.`"

only simple range allowed
The end address of the specified address range is smaller than the start address, e.g. 0xF000--0x3000 is not a simple range.

only ON or OFF allowed
The only acceptable arguments of the current command are ON and OFF.

only ON, OFF, or AUTO allowed
The only acceptable arguments of the current command are ON, OFF, or AUTO.

toggling not allowed in PRACTICE, must supply ON or OFF
A PRACTICE command that expects ON or OFF as argument, was called inside a PRACTICE script without parameter. If such a command is executed from the command line, skipping the argument will cause the option to toggle from ON to OFF and vice versa. This feature is disabled while a PRACTICE script is running.

only High or Low allowed
The only acceptable arguments of the current command are High and Low.

toggling not allowed in PRACTICE, must supply High or Low
A PRACTICE command that expects "High" or "Low" as argument, was called inside a PRACTICE script without parameter. If such a command is executed from the command line, skipping the argument will cause the option to toggle from "High" to "Low" and vice versa. This feature is disabled while a PRACTICE script is running.

keyword expected
The current command lacks one or more keywords.

invalid combination
This keyword is incompatible with a previous keyword.

WARNING: host configuration has changed, may cause unpredictable results
The host in a controller based environment has changed. This can lead to random problems. It is recommended to restart the debug system when the host needs to be changed.

too fast input, some characters were lost
This error can occur when data was lost while transferring data via InterCom or Remote API. Please contact technical support (support@lauterbach.com).

illegal character (<*code*>) for this context

This error can occur when data was lost while transferring data via InterCom or Remote API. Please contact technical support (support@lauterbach.com).

dialog window overflow, too many elements
Host system is out of memory or internal error. Please contact technical support (support@lauterbach.com).

window not found
The window specified for the current command does not exist. **NOTE**: Window names are case sensitive.

no default window exists
No window was specified for the current command and the command tried to access the default window, which does not exist.

panning not allowed, window freeze
Scrolling in frozen windows is not possible, because new data can not be loaded (usually because CPU is running or memory access is not allowed or possible).

unexpected error while accessing file <*filename*>
An unexpected error occurred while reading or writing the specified file (e.g. disk full).

no access to <*filename*>, file locked
The file is currently locked and cannot be accessed.

no access to <*filename*>, no rights
The specified file can not be accessed, because the file access flags do not allow access.

critical error accessing file <*filename*>
The operating system reported a critical error while accessing the specified file. Please make sure that the application has sufficient access rights to access the folder and modify the file. Also make sure that the file is not exclusively used by another application.

syntax error in <filename>
The operating system reported a syntax error while accessing the specified file.

file <*filename*> not found
A file with the specified name does not exist in the current or specified path.

file <*filename*> already exists
The file about to be created already exists in the current or specified path.

file <*filename*> is directory
The specified file name is the name of a directory, while a file is expected by the command.

directory <*filename*> is file
The specified name is the name of a file, while a directory name was expected by the command.

too many files open

The number of open files succeeds the maximum allowed number of files by the operating system. For MS-DOS, at least 10 open files should be allowed in the CONFIG.SYS file.

file <*filename*> too big
The size of this file is larger than allowed. The maximum file size for TRACE32-ICE (SCU) system is limited to 4 GBytes.

no memory for file <*filename*>
The memory of the TRACE32-ICE (SCU) system is full. Deleting information, which is not longer used (e.g. windows, symbols) and retrying the command may help.

wrong access mode to file <*filename*>
Unexpected error. Please contact technical support (support@lauterbach.com).

error in timestamp information for file <*filename*>
Unexpected error. Please contact technical support (support@lauterbach.com).

file type of file <filename> not supported
The file type of the specified file is not supported by this version of TRACE32. The operation cannot be performed.

file version of file <filename> not supported
The version of the specified file is not supported by this version of TRACE32. The operation cannot be performed.

no such page
The **WinPAGE** with the specified name does not exist and can not be selected using **WinPAGE.select**. Use **WinPAGE.Create** to create a page. **NOTE**: Page names are case sensitive.

log not open
The command **LOG.OPEN** must be used to open a log-file.

log already open
A log file is already opened and in use when the command **LOG.OPEN** was called again.

no information about this context
An unknown error occurred. Please contact technical support (support@lauterbach.com).

no filename specified
The current command expects a file name. bot a file name was not specified.

error: wrong file type <*filename*> : <*magicno*>
The file magic doesn't match the actual CPU type or CPU settings. E.g. ElfNoteSection for different CPU type; file byte order doesn't match CPU byte order; file header size is wrong.

warning: debug file does not match executable file <*filename*>

warning: file contains compressed sections (not supported)

error: character (<*code*> H), offset <*offset*>. in file <*filename*> (use DUMP)

Check file format, recompile and try again. If still fails, contact technical support (support@lauterbach.com). A diagnostic hex dump can be made by executing the **DUMP** command.

error: entry near file offset <*offset*>. in file <*filename*> (use DUMP)
(see below)

error: entry near file offset <*offset*>. (offset <*secoffset*> in <*section*>) in file <*filename*> (use DUMP)
(see below)

entry near file offset <*offset*>. in file <*filename*> (use DUMP)
(see below)

entry near file offset <*offset*>. (offset <*secoffset*> in <*section*>) in file <*filename*> (use DUMP)
(see below)

possible compiler bug near offset <*offset*>. in file <*filename*> (use DUMP)
(see below)

symbol <*name*> at offset <*name*> cannot be matched in file <*filename*>
(see below)

error: seek error in file <*filename*>
(see below)

checksum error (<*checksum*>,<*checksum*>) in file <*filename*>
(see below)

invalid address in file
(see below)

error: inconsistency near offset <*offset*>. in file <*filename*>
(see below)

index out of range near offset <*offset*>. in file <*filename*>
(see below)

unexpected end of file
(see below)

inconsistency (<*number*>) in postprocessor
(see below)

wrong location description
(see below)

wrong location description near file offset <*offset*>. (offset <*secoffset*> in <*section*>) in file <*filename*>
(see below)

error: data section near offset <*offset*>.

The contents of the file specified with the **Data.LOAD** command do not match the selected data format of the load command, the debugger detected the wrong file format (if **Data.LOAD.auto** was used), or the contents are corrupted, abnormal or unexpected. Please check

- that the file has not been corrupted while transferring e.g. to another system,
- the specified file is in the intended format (compiler and linker configuration),
- the right data format is selected, e.g. use **Data.LOAD.Elf**,
- the right options are used regarding the compiler, e.g. **Data.LOAD.Elf rom.elf /METROWERKS**,
- if a new compiler version could fix this error,
- if the used compiler is supported by the currently installed debugger software. If the compiler is newer than the debugger software, please upgrade the debugger software and try again.

If the problem can not be fixed, please contact technical support (support@lauterbach.com). When possible, sending the file which caused the problem will help to quickly solve this problem.

convert at address <*address*> in file <*filename*>
The translation from a bit sized file to a byte size one has problems, if not all bits of a byte are valid. In this case the undefined bits are filled with zero and this message is generated.

overlapping mappings in file <*modulname*> : <*affected_address_ranges*>
The debugger detected overlapping address ranges in the specified file. This warning will be displayed if a project used overlays (different code sections are located at the same address and replaced on demand by the application)

overlapping stack frame information <*program_name*> : <*address*>
The debugger detected overlapping address ranges in the specified file. Please check linker configuration.

no format selected
A file format has to be specified for the **Data.SAVE** command, e.g. **Data.SAVE.Binary**.

external reference (<*symbol*>)
When loading the file, external references were found which could not be resolved. Please contact technical support (support@lauterbach.com).

MMU translation wrong for segment <*segname_name*>
The debugger detected an MMU translation problem. Please check linker configuration.

number of function entry / exit points don't match
The debugger detected that `.startfn` and `.endfn` do not match. Check compiler and linker settings.

Bad type indexed, try loading with /GLOBTYPES option
Data.LOAD.Elf had a problem loading the file. There is a chance that the option **/GLOBTYPES** will solve the problem. If the problem is not solved, please contact technical support (support@lauterbach.com).

bit/byte segment type mismatch
The debugger detected a bit/byte segment mismatch. Check compiler and linker settings.

memory overflow

The host ran out of memory when loading the specified file. A corrupted file can also cause this error.

table overflow
An internal overflow occurred. A corrupted file can cause this error. Please contact technical support (support@lauterbach.com).

hash table overflow
An internal overflow occurred. A corrupted file can cause this error. Please contact technical support (support@lauterbach.com).

nesting stack overflow
An internal overflow occurred. A corrupted file can cause this error. Please contact technical support (support@lauterbach.com).

too many elements per record
An internal overflow occurred. A corrupted file can cause this error. Please contact technical support (support@lauterbach.com).

too many sections defined
An internal overflow occurred. A corrupted file can cause this error. Please contact technical support (support@lauterbach.com).

load options don't match with file
One of the options of **Data.LOAD** cannot be used for the file to be loaded.

Illegal source index
The loaded file contains an illegal source index. Check if a new compiler version is available.

Relocatable file
The loaded file is relocatable, e.g. a shared object file or a dynamic link library. The file has to be loaded with its sections relocated using **Data.LOAD** with the /RELOC option.

base address of .debug section is not zero
The base address of the `.debug` section is not zero. This is an atypical case. Please check compiler/linker settings.

explicit PRACTICE macro declaration expected
The preceding **PMACRO.EXPLICIT** command requires that you explicitly declare all subsequent macros with the **LOCAL** or **PRIVATE** command, e.g. `LOCAL &file`, before they can be initialized, e.g. `&file="myfile.txt"`

Alternatively, you can end the explicit macro declaration range with the **PMACRO.IMPLICIT** command.

explicitly declared PRACTICE macro already exists
The preceding **PMACRO.EXPLICIT** command requires that you explicitly declare all subsequent macros exactly at one place with the **LOCAL** or **PRIVATE** command, e.g. `LOCAL &file`.

Alternatively, you can end the explicit macro declaration range with the **PMACRO.IMPLICIT** command.

PRACTICE script not running

The PRACTICE command **CONT** was called without a stopped PRACTICE script. PRACTICE scripts can be stopped using the **STOP** command inside the script, by an error event or using the **STOP** button in the toolbar.

PRACTICE block nesting error in line <*number*>
The PRACTICE script contains a **GOTO** command which attempts to jump into a block in a different nesting level.

PRACTICE syntax error in line <*number*>
The PRACTICE script contains a syntax error at the given line. Use **PLIST** to check the script.

block nesting error
In the currently executed PRACTICE script, a block end ")" was about to be executed without a block start "(".

label must be local
The destination label of the current **GOTO**, **GOSUB** command not in the local frame.

must be a macro name
The argument of a **LOCAL** or **LOCAL** command must be a macro. Macros always start with "&".

e.g. `LOCAL  &my_local_i  &l_filename  &offset`
e.g. `GLOBAL  &project_directory  &subprojectname`

illegal macro name
A macro name contains an illegal character. Only characters "a"-"z", "A"-"Z", "0"-"9", "@" and underscore "_" are allowed. The first character must not be "0"-"9".

illegal macro name (must not start with a numerical digit)

A macro name must not start with a numeric digit. This restriction is valid starting with intermediate build 11/2019 and release version 02/2020.

no such line number
A line number was passed as argument of a **GOTO**, **GOSUB** or **JUMPTO** command is higher than the number of lines of the active PRACTICE script.

no such label
In a PRACTICE SCRIPT, **GOTO** or **GOSUB** was called with an undeclared label. **NOTE**: Labels must start in the first column and must be followed by a colon. Labels are case sensitive.

double defined label
In a PRACTICE script file, a more than one label with the same name were found.

PRACTICE stack overflow
A PRACTICE stack overflow occurs when too many nested calls took place. E.g.:

- A PRACTICE file which recursively calls itself (endlessly)
- The commands executed upon an event (e.g. **ON ERROR**), causes the event to occur again
- The execution of a PRACTICE script file (*.cmm) was interrupted and the PRACTICE stack not cleared. Use **END** commend to clear the PRACTICE stack.

In order to solve a PRACTICE stack overflow, use **PLIST** and **PMACRO** windows to debug the PRACTICE script.

PRACTICE stack underflow
In the currently executed PRACTICE script, a **RETURN** command was to be executed without a preceding **GOSUB** call.

PRACTICE data file not open
The write or read PRACTICE command was called with a file number that is not opened.

PRACTICE file number is too large
The given file number exceeds the limits (0..119).
e.g. `PRINT FILE.EOF(300.)`

PRACTICE pipe not open
An InterCom pipe was about to be read or written, but no pipe with this number was opened.

PRACTICE pipe already open
The PRACTICE command **InterCom.PipeOPEN** was called for an already opened file number.

unknown printer
The printer specified in the **PRinTer.select** command does not exist.

error accessing printer device
Check printer configuration, printer on-line state and cables.

error reading postscript header file
The syntax of the postscript header file is incorrect.

unknown area message window
The **AREA** message window with the specified name does not exist. Create an **AREA** message window using **AREA.Create** *<name>*. **NOTE**: The name is case sensitive.

too many areas
It was attempted to create more than the maximum allowed amount of **AREA** message windows. Close unused windows using **AREA.CLOSE**.

search path too long
The path name specified in the current PRACTICE command is too long.

too many search paths
The maximum number of search paths is exceeded.

directory name as search path expected
Directory name is missing.
Or syntax of deprecated old PATH command is used in combination with new commands.
e.g. PATH.Set + C:\t32\myscriptdirectory

packing to LZW failed

An error occurred while compressing a file using the **PACK** command.

unpacking of LZW failed
An error occurred while decompressing a file using the **UNPACK** command. Check if the file is in compressed format with the corresponding algorithm (ZIP/PACK).

packing to ZIP failed
An error occurred while compressing a file using the **ZIP** command.

unpacking of ZIP failed
An error occurred while decompressing a file using the **UNZIP** command. Check if the file is in compressed format with the corresponding algorithm (ZIP/PACK).

no input area
No **AREA** message window is selected as input source of the **ENTER** command. Use **AREA.Select**. (e.g. `AREA.Select A000` to select the default **AREA** message window)

file is read only
The file to be opened for writing has the file attribute "read-only" set.

editor buffer full, no more files allowed
Only a restricted number of files can be edited at the same time. Close unused files with **EDIT.CLOSE**. All opened files can be displayed using the command **EDIT.List**.

file not in editor
EDIT.SAVE *<filename>* was called with a file name that is currently not opened in the editor. All opened files can be displayed using the command **EDIT.List**.

wrong file format
The format specified with the **Data.LOAD** command does not match the format of the specified file.

unknown file format (use Data.LOAD.Binary to load binary files)
The specified file could not be loaded, because **Data.LOAD.auto** was not able to detect the data format of the file. If the file is a binary file, load it with **Data.LOAD.Binary**. Note that the behavior of **Data.LOAD.auto** changed with version 69130. In the previous implementation, files of unknown format were loaded as binary files.

wrong magic code in file
File magic doesn't match actual CPU type.

illegal floating point format
The loaded file contains an illegal or unknown floating point format. Please contact technical support (support@lauterbach.com).

type-conversion overflow
The given value is too big and cannot be converted.

wrong cycle type
The specified cycle type does not match.

bus sequence not defined

The specified bus sequence is not defined.

sub-processor timeout, contact technical support (support@lauterbach.com)
A hardware or software problem occurred. Please contact technical support (support@lauterbach.com).

sub-processor timeout during boot, contact technical support (support@lauterbach.com)
A hardware or software problem occurred. Please contact technical support (support@lauterbach.com).

sub-processor queue overflow, contact technical support (support@lauterbach.com)
A hardware or software problem occurred. Please contact technical support (support@lauterbach.com).

no symbols available in the selected system
No debug symbols could be found.

cannot resolve InterCom name
An **InterCom** command was entered, but InterCom is not enabled (check configuration file), or the specified host name could not be resolved.

error executing remote InterCom command
The command executed via InterCom caused an error event at the remote host.

no response from InterCom
InterCom is enabled and the host name could be resolved, but the remote host did not response. Check if InterCom is not enabled at the remote host and if the network allows communication between the systems. To adjust the default time out, use the **SETUP.InterComACKTIMEOUT** command.

no additional help available
The error message to be displayed could not be found. Please contact technical support (support@lauterbach.com).

function return value exceeds maximum string length of 4095 bytes

function parameter value too small (><*minvalue*>)

function parameter value too huge (><*maxvalue*>)

internal error : <*error*>
Please contact the manufacturer !

internal error : <*error*> : <*error2*>
Please contact the manufacturer !

window name empty
The given window name is empty.
e.g. `PRINT  WINdow.POSition("","left")`

window name <*name*> not found

The given position item name is empty.
e.g.
```
WINPOS 10. 20.  80. 25.  15. 2. "mylist"
List.auto
PRINT  WINdow.POSition("mylistt","left")
```

item name empty
The given position item name is empty.
e.g.
```
PRINT  WINdow.POSition("mylist","")
```

item name <*name*> wrong
The given window position item name doesn't exist. Please refer function key display of command **WinPOS** for the usable names (LEFT, UP, HSIZE, VSIZE, HSCALE, VSCALE) .
e.g.
```
WINPOS 10. 20.  80. 25.  15. 2. "mylist"
List.auto
PRINT  WINdow.POSition("mylist","leftt")
```

MENU nesting error, check brackets
One of the loaded menu files causes nesting errors. Number of "(" and ")" is not equal.

MENU nesting overflow, too many nested structures
The amount of nested blocks in one of the loaded menu files is too high.

MENU block open not allowed here
A block open "(" is not allowed at this point in the current menu file.

MENU block end not allowed here
A block closing ")" is not allowed at this point in the current menu file.

Block open "(" is expected here
A block open round bracket is missing at this point in the current menu file.

Block open "[" is expected here
A block open square bracket is missing at this point in the current menu file.

Block close ")" is missing
A block close round bracket is missing in the current menu file.

Block close "]" is missing
A block close square bracket is missing in the current menu file.

Block open "(" not expected here
A block open round bracket is not expected here.

Block open "[" not expected here
A block open square bracket is not expected here.

Block close ")" not expected here
A block close round bracket is not expected here.

Block close "]" not expected here

A block close square bracket is not expected here.

Block open bracket expected here
A block open round or square bracket is expected here.

Missing parent menu declaration
The parent menu declaration is missing for this child element.

Parent menu must be of the type MENU or POPUP
The parent menu is expected to be of the type **MENU** or **POPUP**.

Parent menu must be of the type MENU, POPUP or TOOLBAR
The parent menu is expected to be of the type **MENU**, **POPUP**, or **TOOLBAR**.

Parent menu must be of the type TOOLBAR
The parent menu is expected to be of the type **TOOLBAR**.

ELSE without matching IF found
An **ELSE** statement was found without a matching **IF** statement.

TOOLITEM embedded script (..) expected here
An embedded PRACTICE script enclosed by round brackets is expected here for the **TOOLITEM** statement.

TOOLITEM bitmap [..] expected here
An bitmap enclosed by square brackets is expected here for the **TOOLITEM** statement.

too many nested menu definitions
The amount of nested menu definitions in one of the loaded menu files is too high.

Nesting block open missing
A block open "(" is expected at this point.

Nesting block close missing
A block close ")" is expected at this point.

Dialog label not found
The specified dialog label was never declared.

Dialog element at label has wrong type for this operation
The specified dialog element does not support the current operation.

no active dialog
A dialog related command was called without an active dialog.

help search item <*itemstring*> not found
The requested search item doesn't exist in the actual used online help.

Please request a newer online help and/or inform the Lauterbach support about this missing item (support@lauterbach.com).

online help file <*filename*> not found in directory <*manualdirectoryname*>

The required PDF file couldn't be opened from the TRACE32 online help system.

Too many filters defined
The maximum filter number is exceeded

Only one filter is possible
It is not possible to add or delete more than one filter at once. e.g.:
`HELP.FILTER.ADD bdmarm7;bdmmpcp`

Filter expected
The command **HELP.FILTER.Add** was called without specifying a help filter. E.g. in order to add the Linux awareness manual to the help filter list, type `HELP.FILTER.Add rtoslinux`.

Filter expected
The command **HELP.FILTER.Delete** was called without specifying a help filter. E.g. in order to delete the linux awareness manual from the help filter list, type `HELP.FILTER.DELETE rtoslinux`.

active filter expected
The command **HELP.FILTER.Delete** was called with a not active help filter. Only active filter can be deleted.

Value exceeds the range of defined filters
The value is longer than the filter list or negative - value can be only an available filter number to be deleted!

Detection of PDF viewer executable failed! Please select a PDF viewer manually.
The detection of a PDF viewer on the system has failed because no application displaying PDF files was found. Please install a PDF viewer.

No presets found for the selected PDF viewer! Please look up the command line parameters of the selected PDF viewer.
There are no predefined preset values for the detected or selected PDF viewer. Please look up the command line parameters to open PDF files with the selected PDF viewer.

PDF viewer for the HELP system is unconfigured! Use SETUP.PDFViewer to configure a PDF viewer!
There is no PDF viewer configured for the help subsystem to display PDF files. Use **SETUP.PDFViewer** to configure a PDF viewer, or choose **Help** menu > **Setup PDF Viewer**.

Help files are not up to date, see AREA for details
See AREA for a list of old pdf-files, and download the new ones at the Lauterbach server. The update-check can be disabled with **HELP.checkUPDATE OFF**

internal error in TRACE32 online help system

No STP master ID found
High-Level STP decoding requires a master ID to be found in the strace stream. The following measures should be taken if no master ID could be found:

- Verify that STP packets could be received at all: STMxx.List
- For software masters the STM module can output master IDs periodically: STM.MasterRepeat *<value>*
- Assign a hardware master ID to the current trace stream manually: STM.SetMaster *<master_id>*

Error Messages

no logical operator

syntax error in logical operator - character != {AOX}

":" expected

syntax error in logical operator - ending ":" is missing

no arithmetical operator

syntax error in arithmetical operator - character != {AOX}

"#" expected

syntax error in arithmetical operator - ending "#" is missing

no compare operator

syntax error in compare operator - character != {<>=}

```
e.g. 21212!!=123.3
              ^        error position in radixmode decimal
```

In this mode "21212" won't be interpreted as a binary constant.
Rather the decimal constant "21212" will be compared concerning not equal "!=" to the floating point value "=123.3".

error in ASCII constant ("'" expected)

syntax error in ASCII constant - ending "'" is missing
example: "'a" (wrong) - "'a'" (right)

error in string constant ("""" expected)

syntax error in string constant - ending """ is missing

path name expected

path specification is expected - "\" is not enough for a path name

unexpected character in numeric constant

syntax error in numeric constant
character != {numeral|.|hexnumeral|sign|operator|closing bracket|delimiter}character != {0123456789.ABCDEF+-*/<>=:#)}, TAB;NIL}

unexpected character in hex constant

syntax error in hex constant
character != {numeral|hexnumeral|operator|closing bracket|delimiter}
character != {0123456789ABCDEF+-*/<>=:#)}, TAB;NIL}

```
e.g. 22a...54cf
        ^          error position (given decimal point)
     0x1000.0x2000
           ^       error position (missing '.' for range input)
```

unexpected character in integer constant

syntax error in integer constant
character != {0123456789E+-*/<>=:#)}, TAB;NIL}

unexpected character in float constant

syntax error in float constant
character != {0123456789E+-*/<>=:#)}, TAB;NIL}

unknown type specifier '<*character*>' in format string

You have used a type specifier, which is not known by the PRINTF or SPRINTF command, inside the format string. Please consider that you write "%%" inside the format-string, if you just want to print the "%" character. See the documentation of **PRINTF** for more details.

scale factor expected

scale factor expected
character != {0123456789+-}

scale factor expected

scale factor expected
character != {0123456789}

error in float constant

syntax error in float constant
character != {0123456789+-*/<>=:#)}, TAB;NIL}

wrong character in name

this character must not stand in a name

invalid character "<*character*>" in keyword (The characters <*characters*> are not allowed here)

The keyword contains a character which is not allowed here.

"`" expected

the ending "`" escape character is missing
e.g. "name`#" instead of "name`#`" was written

escape characters are used for suppressing of name character syntax check
this is necessary for using reserved characters in names
(all characters which are used for expression operators)
e.g. "#" <-> "#A#"; ":" <-> ":O:"

wrong character in bit or hex mask

this character must not stand in a mask
e.g. "0X1x**.**10!" or "0Xaxx**g**x123"

line or column number 0 does not exist

Line or column number 0 does not exist.
If several HLL commands exist in one source text line, then the column number corresponds to the number of the chosen command (from 1 ascendant numbered).

error in line or column number

syntax error in line or column number
character != {0123456789+-*/<>=:#)}, TAB;NIL}

no digit in binary constant

no digit in binary constant
e.g. "1001 !" or "1001**+**!" instead of "1001!" was written

wrong digit in binary constant

wrong digit in binary constant
e.g.: "0y10**2**" or "10**2**!" (old fashioned)

program name expected

A program name must always come after "\\".

path extension expected

A path extension is expected.
A symbol name "\name" must always come after a "\\programname".

no time unit of measurement

A time unit of measurement is expected.
There are existing 5 kinds of measurement :
"ks" for kiloseconds, "s" for seconds, "ms" for milliseconds,
"us" for microseconds, "ns" for nanoseconds
`e.g. 23.5 milliseconds corresponds to "23.5ms" or "0.0235s" or "23500.us"`

wrong character in hex mask

This character must not stand in a hex mask.
`e.g. "0Xabcxx!"`

wrong character in binary mask

This character '!' must not stand in a binary mask. The new format is mixed with old format for binary mask.
`e.g. "0y01xx01!"`

Please refer for details to chapter **parser changes** too!

wrong character in binary constant

This character '!' must not stand in a binary constant. The new format is mixed with old format for binary constant.
`e.g. "0y011101!"`

Please refer for details to chapter **parser changes** too!

wrong character in hex mask

This character '!' must not stand in a hex mask. The new format is mixed with old format for binary mask.

```
e.g. "0X0axx01!"
```

Please refer for details to chapter **parser changes** too!

wrong character in hex constant

This character '!' must not stand in a hex constant. The new format is mixed with old format for binary constant.

```
e.g. "0X01a101!"
```

Please refer for details to chapter **parser changes** too !

syntax error in numeric constant

syntax error in numeric constant

character != {numeral|.|hexnumeral|sign|operator|closing bracket|delimiter}
character != {0123456789.ABCDEFXY+-*/<>=:#)}, TAB;NIL}

```
e.g.  0z123abc
       ^       error   instead of x for hex constant
```

syntax error in binary constant or mask

syntax error in binary constant or binary mask
character != {binary numeral|mask character}
character != {01X}

```
e.g.  0y 01xx11
        ^       error
```

syntax error in hex constant or mask

syntax error in hex constant or hex mask
character != {hex numeral|mask character}
character != {0123456789X}

```
e.g.  0x 013xxaf
        ^       error
```

syntax error in special filename - closing " is missing

syntax error in special filename
character != "

```
e.g.  \\myprogram\"C:\examples\helloworld.c
                                          ^       error
```

no more symbols expected

At the end of a expression no additional symbols (input characters) are expected with the exception of delimiters.
Generally operators were forgotten to write or delimiter were inserted in the expression erroneous.
`e.g.: "(no+20)i" or "(no+20)* i" instead of "(no+20)*i"`

symbol != {NIL TAB;,}

wrong operand type behind logical operator

operand type after || or :O: <> BOOLEAN

wrong operand type behind logical operator

operand type after || or :O: != RANGE,INTNUMERIC

wrong operand type before logical operator

operand type before || or :O: != BOOLEAN,RANGE,INTNUMERIC,ADDRESS,ADDRESSRANGE

wrong operand type behind logical operator

operand type after ^^ or :X: != BOOLEAN

wrong operand type behind logical operator

operand type after ^^ or :X: != RANGE,INTNUMERIC

wrong operand type before logical operator

operand type before ^^ or :X: != BOOLEAN,RANGE,INTNUMERIC,ADDRESS,ADDRESSRANGE

wrong operand type behind logical operator

operand type after && or :A: != BOOLEAN

wrong operand type behind logical operator

operand type after && or :A: != RANGE,INTNUMERIC

wrong operand type before logical operator

operand type before && or :A: != BOOLEAN,RANGE,INTNUMERIC,ADDRESS,ADDRESSRANGE

wrong operand type behind arithmetical operator

operand type after | or #O# != BINARY,HEX,INTEGER,ASCII,MASK

wrong operand type before arithmetical operator

operand type before | or #O# != BINARY,HEX,INTEGER,ASCII,MASK

wrong operand type behind arithmetical operator

operand type after ^ or #X# != BINARY,HEX,INTEGER,ASCII

wrong operand type before arithmetical operator

operand type before ^ or #X# != BINARY,HEX,INTEGER,ASCII

wrong operand type behind arithmetical operator

operand type after & or #A# != BINARY,HEX,INTEGER,ASCII,MASK

wrong operand type before arithmetical operator

operand type before & or #A# != BINARY,HEX,INTEGER,ASCII,MASK

wrong operand type behind compare operator

operand type after compare != TIME,TIMERANGE

wrong operand type behind compare operator

operand type after compare != NUMERIC,RANGE

wrong operand type behind compare operator

operand type after compare != STRING

error in symbol path

Error in symbol (path/name) input
Reserved for internal usage ! The symbol administration put out the "right" error message !

procedure, variable name, label or line number expected

variable name expected

After path input e.g. "\module\procedure\" a name without path continue character "\" is expected
e.g. "variablename" or "label".

path extension expected

After program name (e.g. "\\Sieve\") another name is expected (e.g. "123", "variablename" or "procedurename").

```
e.g. \\program\module\function
     \\diab\main\123
     \\program\\varname
```

string or name for function parameter expected

String or name for function parameter is expected (e.g. REG("PC") or REG(PC)).The syntax for the name is the same as for symbols. The only restriction is, you can't write any symbol path.

internal error : PAR_224 - <>

internal error : PAR_224
invalid or not implemented address type

wrong operand type before compare operator

operand type before compare != NUMERIC,STRING,RANGE,ADDRESS,ADDRESSRANGE,TIME,
TIMERANGE,BITMASK,BOOLEAN

wrong operand type behind arithmetical operator

operand type after {+-} != NUMERIC,ADDRESS

no negativ string permitted

Operator "-" is invalid before string operand
`e.g.  -"string"`

wrong operand type behind arithmetical operator

operand type after {+-} != NUMERIC,STRING,ADDRESS,TIME

wrong operand type behind arithmetical operator

operand type after {+-} != NUMERIC,ADDRESS

behind string operand no "-" is permitted

Operator "-" is forbidden after string operand.

wrong operand type behind arithmetical operator

operand type after {+-} != ASCII,STRING

wrong operand type before arithmetical operator

operand type before {+-} != NUMERIC,STRING,ADDRESS,TIME

division by zero

Division by zero is forbidden.

wrong operand type behind arithmetical operator

operand type after {*/%} != NUMERIC,ADDRESS

wrong operand type before arithmetical operator

operand type before {*/%} != NUMERIC,ADDRESS

no numeric shift factor

operand type after {<< >>} != NUMERIC

wrong operand type before shift operator

operand type before {<< >>} != BINARY,HEX,INTEGER,ASCII,STRING,RANGE

wrong operator type before logical operand

operator type unary {+-~N#} before BOOLEAN operand

logical not before numeric operand

operator type unary ! or N: before operand type BINARY,HEX,INTEGER,ASCII

not operator before float operand

operator type unary {! ~ N: N#} before operand type FLOAT

wrong operator before range operand

operator type unary {+-~N#} before operand type RANGE

wrong operand type behind operator

operand type after unary {+-!~N:N#} != BOOLEAN,NUMERIC,ASCII,RANGE,ADDRESS, ADDRESSRANGE,TIME

no numeric operand behind range operator

operand type after {++ -- ..} != NUMERIC

wrong operand type before range operator

operand type before {++ -- ..} != NUMERIC,TIME,ADDRESS

error in time constant

e.g.: numeric overflow in time constant

")" expected

Ending ")" is missing. Parenthesis of a expression is incorrect.

operanden types mismatched

wrong operand type combination

only the following combinations of operand types are permitted :
NUMERIC -- NUMERIC, TIME -- TIME, ADDRESS -- ADDRESS,
NUMERIC ++ NUMERIC, TIME ++ TIME, ADDRESS ++ NUMERIC.

wrong operand type behind range operator

operand type after {++ -- ..} != NUMERIC,TIME,ADDRESS

address or numeric operand expected

after {-- ..} address or numeric value expected

access modes mismatched

No different access modes permitted.
Address ranges cannot have different access modes at the same moment.

begin or end address exceeds maximum address

Begin address or begin address + displacement ("beginadr++distance") exceeds maximum address of the CPU.

"}" expected

Ending "}" is missing. Incorrect parenthesis of a expression

operand expected

The symbol which was read is no operand.

```
e.g. "|name", "name #a#0fc" or ")200*name)" or "r(,)"   or
      ^           ^           ^                ^ error positions
e.g. "p:-100"
              ^                                  error position
```

the correct way is "p:(-100)"

numerical displacement expected

After "address++" a numerical displacement is expected.

begin address larger than end address

The begin address cannot be larger than the end address of a address range.

e.g. "20--10" or "0ffffff0++20" (endadr smaller because of numeric overflow) respectively p:1000:5000--222 (offset of endadr must be larger or equal 5000)

internal error : PAR_256

function parameter stack overflow

The internal stack for storing the function parameters is too small.
Too many respectively too long (byte length) function parameters written.

internal error : PAR_258 <>

internal error : PAR_259

internal error : PAR_260

internal error : PAR_261

internal error : PAR_262 <>

no loop variable exist

At the moment no loop variable is available.

function "<*name*>" is not available (locked)

At the moment the function is not available. It is locked, because of e.g. the hardware used from the function does not exist or the emulator is not in the "right" mode.

no function "<*name*>" exists - don't use commands as functions - Press F1 for more details

The given function name is not available or is written wrong.
Please don't mismatch function names with TRACE32 command names.
e.g. print d.l P:0x1000 // gives a SYNTAX error - it's not the same as the line below
print d.l(P:0x1000) // function d.l reads a long value from address P:0x1000
An example for a similar **command** would be
D.IN(D:0x2000) // this command prints the content of address D:0x2000 into the message line

length of the result string is larger than 4095 (255)

symbol path too long

symbol path buffer overflow - path too long

function parameter or ")" expected

After "FUNCNAME(" a function parameter or a closing ")" is expected.

operand or expression required

Within an expression an access mode cannot stand alone.
e.g.: The inputs of "funcname(123,up:)" or "123+ED:" are wrong !

too many intermediate results

internal RESULTSTACK overflow - too many intermediate results
Maybe a solution is the change of the expression structure.

internal error : PAR_271

no actual parameter expected

According to the function definition there are no parameters permitted.

wrong parameter type : expected <expression types>

The written function parameter has a wrong **type**.

no more parameter expected

Actually the function has more parameter (number) than defined.

more parameter required

Actually the function has less parameter (number) than defined.

parameter expected

According to the function definition there are parameters expected.

internal error : PAR_277

address types mismatched <type1> <type2>

Different address types are forbidden.

internal error : PAR_279 - <>

no active access mode table (no device selected or internal error : PAR_280)

At the moment the input of a access mode is locked, because a device without a access mode table respectively no device is selected.
But if the device should have a access mode table, then the internal error PAR_280 was happening.

function stack overflow too many function calls

overflow of the static result stacks

function stack overflow too many function calls

overflow of the code respectively the parameter stack

result is empty-range

A EMPTY range is created. EMPTY ranges must not be created.
`e.g. "1--5&&6--10"`

wrong operand type behind logical operator

operand type after || or :O: != BOOLEAN, RANGE, INTNUMERIC, ADDRESS, ADDRESSRANGE

wrong operand type behind logical operator

operand type after ^^ or :X: != BOOLEAN, RANGE, INTNUMERIC, ADDRESS, ADDRESSRANGE

wrong operand type behind logical operator

operand type after && or :A: != BOOLEAN, RANGE, INTNUMERIC, ADDRESS, ADDRESSRANGE

wrong operand type behind compare operator

operand type after compare != NUMERIC,RANGE,STRING,ADDRESS,ADDRESSRANGE

wrong operand type behind logical operator

operand type after ! or N: != BOOLEAN, RANGE, INTNUMERIC, ADDRESS, ADDRESSRANGE

wrong operand type behind arithmetical operator

operand type after ~ or N# != BINARY,HEX,INTEGER,ASCII

numeric or time operand expected

operand type after + != NUMERIC,TIME

numeric or time operand expected

operand type after - != NUMERIC,TIME

time operand expected

operand type after {++ -- ..} != TIME

invalid access mode or segment not permitted

wrong operand type behind access mode

operand type behind access mode != intvalues,RANGE,ADDRESS,ADDRESSRANGE

access modes mismatched
Different access modes are forbidden.

internal error : PAR_296

too many single address ranges

number exceeds internal range

internal error : PAR_299

symbol path too long

symbol path buffer overflow - too many path changes

address types mismatched

Different address types are forbidden.

16 bit offset expected

the given value for an address offset is larger than 16 bits
`e.g. P:0x1000:0x2000--0x12345`

wrong operand type behind arithmetical operator

operand type after + | - != NUMERIC,ADDRESS

number overflows internal maximum number (integervalue)

number underflows internal minimum number (integervalue)

number exceeds maximum address

wrong operand type behind compare operator

operand type after compare != ADDRESS,ADDRESSRANGE

wrong operand type behind compare operator

operand type after compare != NUMERIC,RANGE,STRING

wrong operand type behind compare operator

operand type after compare != NUMERIC,RANGE,ADDRESS,ADDRESSRANGE

wrong operand type behind compare operator

operand type after compare != STRING,ADDRESS,ADDRESSRANGE

time range begin out of bounds

The time range interval bound must be larger than 0.

time range end out of bounds

2. nd bound > maximum time range - internal bound

time range end < time range begin

in time range intervals the begin time must be larger than end time

too many part ranges

maximum number of range part intervals exceeded

internal error : PAR_315 <>

internal error : PAR_316

wrong operator before address operand

operator type unary {+-N#} before operand type ADDRESS

result is empty address range

An EMPTY address range is created. EMPTY address ranges must not be created.
`e.g. "n:(sp:0)--(sp:0ffffffff)" or "(u:0--0fff):A:(u:2000)"`

internal error : PAR_319 = <>

internal error : PAR_320

too many part address ranges

maximum number of address range part intervals exceeded

wrong operator before address range operand

operator type unary {+-~N#} before operand type ADDRESSRANGE

wrong operand type behind logical operator

operand type after || or :O: != BOOLEAN,RANGE,INTNUMERIC

wrong operand type behind logical operator

operand type after || or :O: != BOOLEAN,ADDRESS,ADDRESSRANGE

wrong operand type behind logical operator

operand type after || or :O: != RANGE,INTNUMERIC,ADDRESS,ADDRESSRANGE

wrong operand type behind logical operator

operand type after || or :O: != ADDRESS,ADDRESSRANGE

internal error : PAR_327 = <>

internal error : PAR_328 = <>

string, numeric constant, name too long

string or numeric constant, name exceeds the maximum length
internal error : PAR_329

wrong operand type behind logical operator

operand type after ^^ or :X: != BOOLEAN,RANGE,INTNUMERIC

wrong operand type behind logical operator

operand type after ^^ or :X: != BOOLEAN,ADDRESS,ADDRESSRANGE

wrong operand type behind logical operator

operand type after ^^ or :X: != RANGE,INTNUMERIC,ADDRESS,ADDRESSRANGE

wrong operand type behind logical operator

operand type after ^^ or :X: != ADDRESS,ADDRESSRANGE

wrong operand type behind logical operator

operand type after && or :A: != BOOLEAN,RANGE,INTNUMERIC

wrong operand type behind logical operator

operand type after && or :A: != BOOLEAN,ADDRESS,ADDRESSRANGE

wrong operand type behind logical operator

operand type after && or :A: != RANGE,INTNUMERIC,ADDRESS,ADDRESSRANGE

wrong operand type behind logical operator

operand type after && or :A: != ADDRESS,ADDRESSRANGE

wrong operator before time operand

operator type unary {~!N#N:} before operand type TIME

access mode combination e:<addressmode> not allowed

The combination of the two access modes isn't allowed, because no dualport access is possible for the second access mode.

internal error : PAR_340

internal error : PAR_341

segment descriptor, LDT, task or bank number mismatched

No different segment descriptors, LDT, task or bank numbers permitted.
Address ranges cannot have different address extensions at the same moment.

internal error : PAR_343 <>

symbol names not allowed

no loop variable "?" allowed

no more symbols expected - superfluous "("

At the end of an expression no additional symbols (input characters) are expected with the exception of delimiters.
Generally operators were forgotten to write or delimiter were inserted in the expression erroneous.

```
e.g.:  "(no+20)i"  or "(no+20)* i"   instead of
       "(no+20)*i" or "10*0a("
```

symbol != {NIL TAB;,}

no more symbols expected - superfluous "[" or "]"

At the end of an expression no additional symbols (input characters) are expected with the exception of delimiters.
Generally operators were forgotten to write or delimiter were inserted in the expression erroneous.

```
e.g.:  "(no+20)i"   or "(no+20)* i"   instead of
       "(no+20)*i"  or "10*0a]"
```

Symbol != {NIL TAB;,}

no more symbols expected - superfluous ")"

At the end of an expression no additional symbols (input characters) are expected with the exception of delimiters.

Generally operators were forgotten to write or delimiter were inserted in the expression erroneous.

```
e.g.:  "(no+20)i" or "(no+20)* i"   instead of
       "(no+20)*i"   or   "10*0a)"
```

symbol != {NIL TAB;,}

combination of A: access mode and segment address forbidden

A combination of a physical access mode like AP: and a segment address isn't allowed.

```
e.g.  Data.List  AP:0x2:0x34567890
```

The only exception is a PPC CPU. In this case a 4 bit segment address is allowed as an address extension of the upper 36 physical address bits.

unexpected access mode

The symbol which was read is no correct operand. The symbol was interpreted as an access mode and at these position inside of the expression no access mode is expected. Maybe you've just inserted some spaces.

```
e.g. "main:o :func1", "main:o:sp:1000"
      ^                       ^                  error positions
```

wrong operand type behind compare operator

operand type after compare != ADDRESS,ADDRESSRANGE,TIME,TIMERANGE

wrong operand type behind compare operator

operand type after compare != STRING,TIME,TIMERANGE

wrong operand type behind compare operator

operand type after compare != STRING,ADDRESS,ADDRESSRANGE,TIME,TIMERANGE

wrong operand type behind compare operator

operand type after compare != NUMERIC,RANGE,TIME,TIMERANGE

wrong operand type behind compare operator

operand type after compare != NUMERIC,RANGE,ADDRESS,ADDRESSRANGE,TIME,TIMERANGE

wrong operand type behind compare operator

operand type after compare != NUMERIC,RANGE,STRING,TIME,TIMERANGE

wrong operand type behind compare operator

operand type after compare != NUMERIC,RANGE,STRING,ADDRESS,ADDRESSRANGE,TIME,

TIMERANGE

dress exceeds 16 bit value

only 16 bit segment addresses are supported

offset exceeds 16 bit value

only 16 bit offsets are supported

wrong operand type behind segment

operand type after segment != NUMERIC,RANGE
```
e.g.: "1000:(22x)"
```

offset expected - wrong operand type or symbol behind segment

symbol or operand type after segment != NUMERIC
```
e.g.: "1000:22s", "1000::o:20", "1000:y.end(func0)"
```

constant, (expression) or function expected - the given symbol isn't an offset
```
e.g. p:1000:articlenumber   p:1000:ip                    wrong
            ^                       ^    p:1000:reg(ip)  correct
e.g. Tree::a:vfunc3   "Tree::a:" will be treated as a segment name
             ^ wrong
e.g. 10:2000:vfunc3:1234  "vfunc3" will be treated as a segment name
```

no segment or access mode <mode> exist

the given name for segment or access mode doesn't exist

float numbers not allowed

the input of float numbers is locked at the moment

internal error : PAR_365 : <>

internal error : PAR_365
invalid or not implemented address type

internal error : PAR_366 : <> : <>

internal error : PAR_366
Please contact the manufacturer !

access mode combination a:<mode> not allowed

The combination of the two access modes isn't allowed, because no absolute address is possible for the second access mode.

internal error : PAR_368 : <> : <>

internal error : PAR_368
invalid result type or not implemented address type

symbol name or path is too long

Symbol name respectively the symbol path for function parameter are too long (e.g. REG("PC") or REG(PC)).
The syntax for the name is the same as for symbols. The only restriction is the name length limit of currently 4095 (255) characters.

wrong operand type behind arithmetical operator

operand type after {+-} != TIME

wrong operand type behind arithmetical operator

operand type after {+-} != NUMERIC,ADDRESS,TIME

numeric overflow in time expression

constant for byte address expected

A hex constant is expected after the bit access mode.
```
e.g. bit:1011!.1    bit:counter.9    bit:\\prog\func1\i
```

Labels for byte addresses are used with access mode.
```
e.g. bit:com1.rts
```

byte address too big

The hex constant exceeds the maximum byte address of 0fffffff.
```
e.g. bit:10000000.1
```

hex constant or symbol as bitaddress expected

A hex constant is expected after the byte address.
```
e.g. bit:100.101!   bit:counter.'a'   com1.1ax   com1.0.1   (typical errors)
     bit:100..3
```

bit number too big

The hex constant exceeds the maximum bit number of 15 (decimal).
```
e.g. bit:100.10
```

internal error : PAR_377 : <> : <>

internal error : PAR_377
Please contact the manufacturer !

unexpected segment

The combination absolute access mode and segment or task number isn't permitted.

```
e.g.  ap:1000:200      a:bank1:200
        ^               ^                  error positions
```

banknumber, space ID, LDT or tasknumber exceeds 16 bit value

only 16 bit bank numbers, space IDs, LDTs or tasknumbers are supported

```
e.g. SPS:0x12345:30:0x2000   instead of   SPS:0x1234:30:0x2000
```

unexpected banknumber

A banknumber and a segment address were given, but the actual CPU doesn't support it or respectively it's a syntax error (2 segment addresses recognized).

```
e.g. 1000:2000:30   instead of   1000:2000:o:30
```

unexpected NIL in HLL expression - often only closing ')' or "" missing

In some cases is only a closing ')' or "" expected !

internal error : PAR_382

internal error : PAR_382

Please contact the manufacturer !

"" expected

Unexpected NIL in HLL expression is recognized. Normally a closing "" is missing in a function parameter.

```
e.g. print data.string("statusmessage)
                                    ^  error
```

')' expected

Unexpected NIL in HLL expression is recognized. Normally a closing ')' is missing after a function parameter.

```
e.g. print data.string((statusmessage)
                                     ^  error
```

oldfashioned operator and operand locked in current radix mode

In the current parser mode is the old syntax of operators and operands locked. Please switch mode to CLASSIC (**SETUP.RADIX CLASSIC**) or use new syntax.

```
old   syntax operators:  N:  :A:  :X:  :O:  N#  #A#  #X#  #O#  <>  ><  =<  =>
new syntax operators:  !  &&  ^^  ||  ~    &    ^    |    !=  !=  <=  >=
old   syntax operands:    10101!     1000    101xx1!      1axxd
new syntax operands:   0y10101   0x1000   0y101xx1   0x1axxd
e.g. print  N:10af
            ^  error
     print  !0x10af
            ^  ok
```

Please refer for details to chapter **parser changes** too !

second '.' expected

single '.' not allowed as range operator - normally second '.' forgotten

```
e.g. break.set P:0x100.0x200     ; wrong
                      ^          ; error position
     break.set P:0x100..0x200    ; correct - addressrange from 100 to 200
     break.set .sieve            ; wrong
               ^                 ; errorposition
     break.set sieve             ; correct with command "sYmbol.PREFIX ."
     break.set `.sieve`          ; correct
```

Please refer for details to chapter **parser changes** too !

no more symbols expected - superfluous "@"

At the end of an expression no additional symbols (input characters) are expected with the exception of delimiters.

Generally operators were forgotten to write or delimiter were inserted in the expression erroneous.

```
e.g.:  "(no+20) @l"   instead of   "(no+20)@l"
```

symbol != {NIL TAB;,}

wrong compare operator before or behind BITMASK

compare operator before or behind BITMASK != "==" or "!="

```
e.g.  r(pc)<=0Y1010xxxx
           ^             ; error position
      r(pc)==0Y1010xxxx  ; OK
```

wrong operand type behind compare operator

operand type after compare operator != NUMERIC,RANGE,BITMASK

wrong operand type after compare operator

operand type after compare operator != INTNUMERIC
Only numeric operands could be combined with bitmasks.

```
e.g.  0Y0111xxxx!=10.ns     ; wrong
                 ^          ; error
      0Y0111xxxx!=0X12000   ; OK
```

wrong operand type behind compare operator

operand type after compare operator != INTNUMERIC,TIME,TIMERANGE

wrong operand type behind compare operator

operand type after compare operator != INTNUMERIC,ADDRESS,ADDRESSRANGE

wrong operand type behind compare operator

operand type after compare != INTNUMERIC,ADDRESS,ADDRESSRANGE,TIME,TIMERANGE

wrong operand type behind compare operator

operand type after compare operator != INTNUMERIC,STRING

wrong operand type behind compare operator

operand type after compare operator != INTNUMERIC,STRING,TIME,TIMERANGE

wrong operand type behind compare operator

operand type after compare != INTNUMERIC,STRING,ADDRESS,ADDRESSRANGE

wrong operand type behind compare operator

operand type after compare != INTNUMERIC,STRING,ADDRESS,ADDRESSRANGE,TIME,TIMERANGE

internal error : PAR_398

String function return value is too long (>4096 bytes).
The string return value of the PRACTICE function exceeds the internal maximum size.

segment register number exceeds 4 bit value

Only 4 bit segment register numbers (<= 0x0f) are allowed for PPC addresses.

modulo operation not permitted with time values

The modulo operator "%" after the time value isn't permitted.

```
e.g.  134.50ns%20.
              ^             ; error position
```

no multiplication of 2 time values permitted

The multiplication operator "*" between the 2 time values isn't permitted.

```
e.g.  134.50ns*200.s
              ^             ; error position
```

wrong operand type behind multiplication operator

operand type after * != NUMERIC

```
e.g. 200.ns*12.3s
            ^             ; error position
```

wrong operand type behind division operator

operand type after / != NUMERIC,TIME

```
e.g. 200.ns/P:0x100
            ^             ; error position
```

wrong operand type behind division or modulo operator

operand type after { / % } != NUMERIC
time values aren't permitted

```
e.g. 12.345%200.ns
            ^               ; error position
     0x2fff/250.ns
            ^               ; error position
```

wrong operand type behind arithmetical operator

operand type after * != NUMERIC,ADDRESS,TIME

wrong operand type behind arithmetical operator

operand type after {*/%} != NUMERIC,ADDRESS

wrong operand type before arithmetical operator

operand type before {*/} != NUMERIC,ADDRESS,TIME

wrong compare operator before or behind BOOLEAN

compare operator before or behind boolean value != "==" or "!="

```
e.g.   IF  ICE()<=("&hw_selected"=="")
               ^                        ; error position
       IF  ICE()==("&hw_selected"=="")  ; OK
```

wrong operand type behind compare operator

operand type after compare operator != BOOLEAN

wrong operand type behind compare operator

operand type after compare != NUMERIC,RANGE,STRING,ADDRESS,ADDRESSRANGE,TIME,

TIMERANGE,BOOLEAN

wrong operand type behind compare operator

operand type after compare != NUMERIC,RANGE,STRING,ADDRESS,ADDRESSRANGE,BOOLEAN

wrong operand type behind compare operator

operand type after compare != NUMERIC,RANGE,STRING,TIME,TIMERANGE,BOOLEAN

wrong operand type behind compare operator

operand type after compare != NUMERIC,RANGE,STRING,BOOLEAN

wrong operand type behind compare operator

operand type after compare != NUMERIC,RANGE,ADDRESS,ADDRESSRANGE,TIME,TIMERANGE,BOOLEAN

wrong operand type behind compare operator

operand type after compare != NUMERIC,RANGE,ADDRESS,ADDRESSRANGE,BOOLEAN

wrong operand type behind compare operator

operand type after compare != NUMERIC,RANGE,TIME,TIMERANGE,BOOLEAN

wrong operand type behind compare operator

operand type after compare operator != NUMERIC,RANGE,BITMASK,BOOLEAN

wrong operand type behind compare operator

operand type after compare != INTNUMERIC,STRING,ADDRESS,ADDRESSRANGE,TIME,TIMERANGE,BOOLEAN

wrong operand type behind compare operator

operand type after compare != STRING,ADDRESS,ADDRESSRANGE,TIME,TIMERANGE,BOOLEAN

wrong operand type behind compare operator

operand type after compare != INTNUMERIC,STRING,ADDRESS,ADDRESSRANGE,BOOLEAN

wrong operand type behind compare operator

operand type after compare != STRING,ADDRESS,ADDRESSRANGE,BOOLEAN

wrong operand type behind compare operator

operand type after compare operator != INTNUMERIC,STRING,TIME,TIMERANGE,BOOLEAN

wrong operand type behind compare operator

operand type after compare != STRING,TIME,TIMERANGE,BOOLEAN

wrong operand type behind compare operator

operand type after compare operator != INTNUMERIC,STRING,BOOLEAN

wrong operand type behind compare operator

operand type after compare != STRING,BOOLEAN

wrong operand type behind compare operator

operand type after compare != INTNUMERIC,ADDRESS,ADDRESSRANGE,TIME,TIMERANGE, BOOLEAN

wrong operand type behind compare operator

operand type after compare != ADDRESS,ADDRESSRANGE,TIME,TIMERANGE,BOOLEAN

wrong operand type behind compare operator

operand type after compare operator != INTNUMERIC,ADDRESS,ADDRESSRANGE,BOOLEAN

wrong operand type behind compare operator

operand type after compare != ADDRESS,ADDRESSRANGE,BOOLEAN

wrong operand type behind compare operator

operand type after compare operator != INTNUMERIC,TIME,TIMERANGE,BOOLEAN

wrong operand type behind compare operator

operand type after compare != TIME,TIMERANGE,BOOLEAN

wrong operand type after compare operator

operand type after compare operator != INTNUMERIC,BOOLEAN

internal error : PAR_433

Warning: in mode RADIX.Classic "N:" is always interpreted as logical NOT (only for ARM)

Please use a more specific access mode instead e.g. **NP: NSP: ND: NUD:** ... or activate a different radix mode like **RADIX.Hex** instead.

In radixmodes hex and decimal **"N:"** will be aways interpreted as access mode **"N:"**.
No usage of old-fashioned logical operator N: is possible.

no more symbols expected - superfluous "*"

At the end of an expression no additional symbols (input characters) are expected with the exception of delimiters.
Generally operators were forgotten to write or delimiter were inserted in the expression erroneous.
`e.g.: "(no+20) *l" instead of "(no+20)*l"`

symbol != {NIL TAB;,}

linenumber or linenumber path expected

After special filename a linenumber is expected only.

```
e.g.:        Data.List \\my_program\"C:\example\helloworld.c"\*123
instead of   Data.List \\my_program\"C:\example\helloworld.c"\123
```

column number expected

After special filename with linenumber path a column number is expected only.

```
e.g.:        Data.List \\my_program\"C:\example\helloworld.c"\123\*10
instead of   Data.List \\my_program\"C:\example\helloworld.c"\123\10
```

no more symbols expected - superfluous ":"

At the end of an expression no additional symbols (input characters) are expected with the exception of delimiters.

Generally operators were forgotten to write or delimiter were inserted in the expression erroneous.

```
e.g.:  "(0x1000) :1"   instead of   "0x1000:1"
```

symbol != {NIL TAB;,}

segment register number exceeds 8 bit value

Only 8 bit segment register numbers (<= 0xff) are allowed for ARM addresses.

address extension bit number exceeds 10 bit value

Only 10 bit address extension values (<= 0x3ff) are allowed for PowerPC addresses.

string function return value is too long (>4096 bytes)

The string return value of the PRACTICE function exceeds the internal maximum size and was be truncated.

segment address exceeds 32 bit value

only 32 bit segment addresses are supported

32 bit offset expected

the given value for an address offset is larger than 32 bits

```
e.g. P:0x1000:0x2000--0x123456789
```

unexpected machine ID

A machine ID was used, but the actual CPU doesn't support it.

```
e.g. P:0x3:::0x2000
```

unexpected space ID

A space ID was used, but the actual CPU doesn't support it.

```
e.g. P:0x2::0x2233
```

machine ID not activated

A machine ID was used, but the actual CPU option isn't set.
For more information, please refer to **SYStem.Option MACHINESPACES**.
`e.g. P:0x3:::0x2000`

space ID not activated

A space ID was used, but the actual CPU option isn't set.
Please refer to **SYStem.Option MMUSPACES** for more information.
`e.g. P:0x2::0x2233`

machine ID <0xnn> not in allowed range 0x00..0x1e

Usually only values 0x00..0x1e are expected for machine IDs.
`e.g. P:0x2345678:::0x1122`

space ID <0xnnnn> not in allowed range of 0x0000..0xffff

Usually only values 0x0000..0xffff are expected for space IDs.
`e.g. P:0x12345::0x4455`

wrong operand type behind machine ID

After machine ID only a space ID or segment or offset is expected.
```
e.g. 0x4:::0x88xx11
     0x4:::2ms
```

wrong operand type behind space ID

After space ID only an offset is expected.
```
e.g. 0x33::0x88xx11     (bitmask)
     0x33::2ms          (time value)
     0x33:: 0x12        (superfluous blank " ")
```

machine name expected

machine name must always come after "\\\".

machine name isn't usable without symbol

access class missing for address value

An access class is missing for a complete address value specification.
`e.g. 0x1000 instead of AXI:0x1000`

this mask constant is maybe interpreted wrong

The actual parsed mask constant is maybe written in old-fashioned format.

This could end in a wrong interpreted value with the new parser version.

```
e.g. data.w0 check_values  0xx33       old-fashioned syntax
     data.w0 check_values 0Xxx33       new-fashioned syntax
     0xx33 ==> xx33 (old parser versions <=V1.90)
     0xx33 ==> 0x33 (new parser versions >=V2.00)
```

Please refer for details to chapter **parser changes** too !

expression expected

the expression has a wrong result type

internal error : STDPAR_102

value too small

value too big

internal error : STDPAR_105

Error Messages Related to the Peripheral View (PER)

overlapping bitfield definition
The parser detected overlapping bit fields in a group.

no base address specified for that group
Specify base address using the **BASE** command, e.g. `BASE iobase()`

group definition must be empty for copy command
If a group contains the **COPY** command, not other group elements are allowed.

no group defined
Group elements were found without a preceding **GROUP** definition.

hidden elements only possible in hidden groups
Hidden group elements were found in a non-hidden group.

element is not part of a SGROUP
A sequence command (CONSTX,GETX,SET,SETX,WRITEBACK) was found outside a sequence group (**SGROUP**).

out statement already used in this group
Only one OUT command is allowed per group.

byte out of group
The byte addressed in this group is out of the defined address range of the group.

text too long for that field
The text of this field is longer than possible with the current setup.

Too few items listed with BIT or BITFLD statement

Too many items listed with BIT or BITFLD statement
The peripheral view file contains a bitfld command, for which more elements are defined than the bit field range allows. E.g. a bit field of 2 bits can not address more than 4 elements.

no line defined for that bit
A PER statement was used inside a group without preceding line command.

tree nesting error
Trees need to begin with **TREE** (or **TREE.OPEN**) and end with **TREE.END**. Either a **TREE/TREE.OPEN** or **TREE.END** is missing.

nesting error
The structure of a **conditional GROUP display** (IF/ELIF/ELSE/ENDIF) or for **conditional interpretation** (SIF/ELSE/ENDIF) is not correct, e.g. ENDIF missing.

group alignment error
The address range of the GROUP/WGROUP/RGROUP statement is not aligned to the access width.

correct FLASH programming timing cannot be achieved
The timing measurement performed by the debugger was out or range, or the flash timing register could not be written by the debugger. On some processors the register is write-once. Please make sure that the flash timing register is writable by the debugger.

FLASH programming error around address <*address*>
Programming Flash at the specified address, or close to this address failed. Possible causes for this errors are:

- Flash sector locked
- Flash is protected by write protect pin
- Flash programming voltage is out of allowed range or disabled
- It was attempted to clear an already programmed bit
- Security mechanism on the board blocks write enable, programming voltage etc.

FLASH algorithm did not execute completely
The flash programming code was not executed until the expected address. Possible causes are interrupts, e.g. watchdog.

FLASH erase error
The Flash device could not be erased. See **Flash programming error** for possible reasons.

cannot access FLASH control registers
Internal error. Please contact technical support (support@lauterbach.com).

Flash data bus bytes are swapped, use target controlled flash programming
The Flash device detected with CFI is connected in reversed byte order. TRACE32 does not support tool based Flash programming for this configuration. Use target controlled flash programming, see **FLASH.CFI** for more information.

cannot detect CFI query information, please check AREA window
Additional error information is printed to **AREA** window.

cannot access on-chip RAM control registers
Internal error. Please contact technical support (support@lauterbach.com).

invalid peripheral register configuration for programming
A register which needs to be modified by the debugger for flash programming (e.g. SYPCR) is write protected, i.e. this write-once register has already been written.

cannot make bulk erase with this device, must be erased sector by sector
The selected device doesn't support bulk erase. Each sector must be erased by a separate **FLASH.Erase** command.

cannot make sector erase with this device
The selected device doesn't support sector erase. The device has to be erased completely using **FLASH.Erase <unitnr>, FLASH.Erase ALL or FLASH.ERASE <full address range>**

parameters for target programming not set (use FLASH.TARGET command)
FLASH.Create was called for target controlled Flash programming, but target controlled Flash programming was not configured. Call **FLASH.TARGET** for configuration.

no flash programming voltage
The Flash programming voltage is out of allowed range or disabled.

no such flash device
A Flash programming command was called with an address or address range for which no Flash device was declared, or with an undeclared flash unit number.

flash sector alignment error
Declared address range is not a multiple of sector size. Check end address of address range and sector size.

memory access class not allowed for flash programming
Memory access class is referencing special memory. Please set up memory access class referencing flash data or program memory.

memory address not supported by flash programming algorithm
The flash algorithm does not support the specified address. Please contact technical support (support@lauterbach.com) to request an update.

Flash programming error
The flash device could not be programmed.

Flash erasing error
The flash device could not be erased.

Flash unlocking error
The flash device could not be unlocked.

Flash locking error
The flash device could not be locked.

Flash alignment error
Some CPUs can access the flashes or flash controllers by a specific aligned addresses (e.g. word, long, 512, 2048 ..). Check the end or start address of the **FLASHFILE** command.

Flash time-out error
The execution of the **FLASHFILE** command exceeded the maximum time-out.

Flash ECC programming error
The flash device failed to program the ECC codes.

Flash protected error
The current flash is protected, so the algorithm file cannot erase/write the flashes.

Flash locked error
The current flash is locked, so the algorithm file cannot erase/write the flashes.

Flashfile zip-loading error
Writing the zip file to the flash device failed.

Flash MMC/SD command error
A notice is displayed in the **AREA** window, informing you which command of the MMC/SD card failed.

Flashfile function not implemented

Flash reading error
The flash device could not be read.

Flashfile BSDL programming error (fail)

TPU access time-out error
The access to the TPU debug registers failed. Check operation mode (TEST) and processor chip.

Cannot execute GO in non-recoverable state
The processor was stopped during non-recoverable state. On this architecture, stopping in non-recoverable state causes loss of the return address. Therefore resuming with **Go** is not possible.

TPU is running
It was attempted to perform a single step or Go while the TPU was already running.

TPU is already stopped
It was attempted to perform a Break while the TPU was already stopped.

TPU is in idle state
Execution of command is not possible, if the TPU is in IDLE state. Start the TPU and run till the TPU is not idle.

PCP is running
It was attempted to perform a single step or Go while the PCP was already running.

PCP is already stopped
It was attempted to perform a Break while the PCP was already stopped.

PCP is in idle state
Execution of commands is not possible, if the PCP is in IDLE state. Trigger the PCP to run it.

WARNING: PCP code at breakpoint has changed, breakpoint removed
The program code of the PCP has been changed by the application

TPU is not responding to PIR command
Internal error. Please contact technical support (support@lauterbach.com).

TPU is not serving any channel
Internal error. Please contact technical support (support@lauterbach.com).

HiPerLoad not available on host
Please check if HiPerLoad is enabled in the TRACE32 start-up configuration file (usually config.t32). The configuration file entry is e.g.

```
HPERLOAD=NETLINK
PORT=20000
```

cannot resolve internet address for HiPerLoad
The internet address used for HiPerLoad could not be resolved. Please check if the firewall is configured properly.

starting UDP socket for HiPerLoad fails on host
The UDP socked specified for use with HiPerLoad could not be started. Please use a socket address which is not used by any other programs on the host.

target not responding to HiPerLoad packets
The target did not respond to the HiPerLoad packets. Please check if the HiPerLoad application on the target is running properly, the target is connected to Ethernet and the firewall allows both TRACE32 and target communicate.

FDX buffer error
The FDX buffer is corrupted. The FDX channel is disabled after this error.

FDX format error
The format of FDX trace data does not match. FDX tracing is stopped after this error.

FDX buffer overflow
The FDX client on the host caused a buffer overflow.

No such FDX
No such FDX channel exists. Channel window must be open for any FDX operation.

too many terminal windows
Only up to nine **TERM.view** window can be opened at a time

TERM.GATE not possible with with method
The selected method does not support full binary transfer.

TERM.GATE protocol fail
The transfer protocol was violated (block length or CRC error).

terminal window with this configuration already open
A **TERM.view** window assigned to the specified terminal address is already opened.

No such terminal
A **TERM** command was called which needs an open terminal window for operation. Configure the terminal via **TERM.METHOD** and open it with **TERM.view**.

invalid magic code or wrong file
The RTOS support file specified with **TASK.CONFIG** is invalid or corrupted. Please re-install the file from CD.

the current task cannot be stopped
The emulator waited until the task becomes active, but the task was not entered within the maximum timeout. Please check if the task and the operating system are running properly.

program definition error
The RTOS support file specified with **TASK.CONFIG** is invalid or corrupted. Please re-install the file from the DVD.

undefined keyword
The used keyword (usually an argument of the current command) is written wrong or is locked.

unknown task name or ID
A command was called with an unknown task name or task ID. Check the **TASK.List.tasks** window for a list of running tasks.

task function usage error: <*description*>
The function defined by the **TASK.CONFIG** command has reported this error. Check the description for the task specific commands.

no way to implement this functionality
A task-related breakpoint could not be set, because the current task magic could not be determined.

Syntax error in ORTI file at line <*line_number*>
The ORTI file is probably corrupted. Please generate new ORTI file and try again. If the problem persists, please contact technical support (support@lauterbach.com).

Syntax error in ORTI expression for <*item*>; symbols loaded?
The ORTI file contains references for *<item>* that could not be resolved. Please check if the application symbols are loaded and if the ORTI file matches the application. If the problem persists, please contact technical support (support@lauterbach.com).

Unexpected end of ORTI file
The ORTI file is probably corrupted. Please generate new ORTI file and try again. If the problem persists, please contact technical support (support@lauterbach.com).

No such task
The task (name, ID or magic number) specified in the current command does not exist. Check the **TASK.List.tasks** window for a list of running tasks.

No such space
The space or process (name, ID or magic number) specified in the current command does not exist. Check the **TASK.List.SPACES** window for a list of existing spaces.

No such machine
The machine (name, ID or magic number) specified in the current command does not exist. Check the **TASK.List.MACHINES** window for a list of existing machines.

function not defined in task program
The function to be executed using the TASK command is not defined. Please contact technical support (support@lauterbach.com).

no default loading agent available
There is no standard target agent for this core (or configuration) available.

loading agent too large for reserved space
The Differential Download agent's program size is larger than the program address space available for use by the agent.

If you did not specify a dedicated address range for the agent, it will automatically be placed behind the data block you download or check. If there is not enough space left to the end of the address range, you must explicitly specify a valid program address range that is large for the target agent to be downloaded.

Load checksum fail between *<address>* and *<address>*
After a differential download completed, differences between the downloaded data and the target memory contents were found. Please check if the differential download agent is running properly and if the tagret memory address ranges are write- and readable by the download agent.

Differential load too complex
The file to be downloaded via differential load contains too many address ranges. Please split the download into several parts or try to reduce complexity or the data file. Another option is to load the data to virtual memory (`Data.LOAD <filename> VM:`) first and then copy from virtual memory to target memory (`Data.COPY VM:<address-range> <target start-address>`)

Target agent verification error
The target agent code has been written to the memory but it's verification has failed. Please check that the memory range used for the target agent is writeable and allows code execution. Ideally, this should be cached memory. By default, this memory range is placed directly behind the downloaded data. If there is no or not enough writeable memory, you can specify an alternative target agent memory range in the command (`Data.LOAD <filename> /DIFFLOAD [<address_range>]`).

Illegal address specification
The target agent memory range must be specified as logical address, not as physical address.

Collision of data and target address space
The specified address ranges of the download data and the target agent code collide with each other. Please select a different memory range for the target agent. Ensure that it is writeable and allows code execution. Ideally, this should be cached memory.

Error Messages Related to Breakpoints

combination of Read/Write and Program not allowed
When software breakpoints are enabled, the combination of Read/Write and Program is not allowed. The default can be changed by the command **SETUP.BREAKDEF**.

Only one advanced breakpoint (condition/command/spot) possible
The amount of advanced breakpoints that can be set at a time depends on the breakpoint capabilities. If the debug system can not determine which breakpoint is hit, more than one advanced breakpoint is only possible if all set breakpoints of the same type also share the same condition.

breakpoint would require stop and resume operation - which is not allowed
Inexact breakpoints or breakpoints with condition may behave like spot points and require a stop and resume operation. **SYStem.CpuSpot** must be set to **Enable** to avoid this error message.

spot breakpoint set - which is not allowed
SYStem.CpuSpot must be set to **Enable** to avoid this error message.

stopping breakpoint set - which is not allowed
SYStem.CpuBreak must be set to **Enable** to avoid this error message.

breakpoint configuration invalid
The current combination of hardware or software breakpoints is not supported. Open **Break.List** window and delete unneeded breakpoints. Please check **Processor Architecture Manual** and reference manual of the processor.

illegal code at software breakpoint
Software breakpoints can not be set certain instructions.

software breakpoints on range not possible
A program breakpoint range made of software breakpoints is not allowed. Use single breakpoints or onchip breakpoint ranges.

illegal address for software breakpoint
An software breakpoint was about to be set to an address which is not allowed due to limitations of the processor. Please check **Processor Architecture Manual** and reference manual of the processor.

no such software breakpoint
Software breakpoints can only stop or spot the target. Other actions are not possible.

no hardware breakpoint available
Hardware breakpoints are not available in this configuration. Hardware breakpoints are implemented by external debugger hardware.

too many different hardware breakpoints
The amount of different hardware breakpoints exceeds the maximum amount provided by the debug hardware. Please check **Processor Architecture Manual** for details.

no such hardware breakpoint available
The hardware breakpoint about to be set is not provided by the debug hardware. Please check **Processor Architecture Manual** for details.

too many hardware breakpoints set
The amount of hardware breakpoints exceeds the maximum amount provided by the debug hardware. Please check **Processor Architecture Manual** for details.

data not allowed for this hardware breakpoint
The selected hardware breakpoint does not support data value comparison. Please check **Processor Architecture Manual** for details.

address does not fit in hardware breakpoint resource
The address or address range of the current breakpoint does not fit into the debug hardware's breakpoint logic. Please check if the address or address range meets the alignment/range requirements of the debug hardware (e.g. if the breakpoint consists of an address value and address mask register). See **Processor Architecture Manual** for details.

data does not fit in hardware breakpoint resource
The data value assigned to the hardware breakpoint does not meet the data format provided by the breakpoint resource on the debug hardware, e.g. if the debug hardware provides a value and mask register. Please check **Processor Architecture Manual** for details.

no hardware breakpoint of this type possible
A hardware breakpoint of the desired type is not provided by the debug hardware. Please check **Processor Architecture Manual** for details.

hardware breakpoint resource sharing conflict
The hardware breakpoint about to be set uses breakpoint resources of the debug hardware which are already used by other breakpoints. Open **Break.List** window and delete unneeded breakpoints. Please check **Processor Architecture Manual** for details.

Cannot map FLAG when hardware breakpoints are set
There is a resource sharing conflict between a hardware breakpoint and a mapped FLAG. Unmap FLAG or disable hardware breakpoint.

Cannot set hardware breakpoint when FLAG is mapped
There is a resource sharing conflict between a hardware breakpoint and a mapped FLAG. Unmap FLAG or disable hardware breakpoint.

no on-chip breakpoints available
On-chip breakpoints are not provided by the processor. Please use software breakpoints.

no such on-chip breakpoint available
The desired on-chip breakpoint is not provided by the processor. Please check **Processor Architecture Manual** and reference manual of the processor.

too many on-chip breakpoints set
More on-chip breakpoints are set than provided by the processor. Please check **Processor Architecture Manual** and reference manual of the processor. Open **Break.List** window and delete unneeded breakpoints.

too many on-chip program breakpoints set

More on-chip breakpoints are set than provided by the processor. Please check **Processor Architecture Manual** and reference manual of the processor. Open **Break.List** window and delete unneeded breakpoints.

too many on-chip read/write breakpoints set
More on-chip breakpoints are set than provided by the processor. Please check **Processor Architecture Manual** and reference manual of the processor. Open **Break.List** window and delete unneeded breakpoints.

too many on-chip context breakpoints set
More on-chip breakpoints are set than provided by the processor. Please check **Processor Architecture Manual** and reference manual of the processor. Open **Break.List** window and delete unneeded breakpoints.

too many on-chip machine breakpoints set
More on-chip breakpoints are set than provided by the processor. Please check **Processor Architecture Manual** and reference manual of the processor. Open **Break.List** window and delete unneeded breakpoints.

data not allowed for this on-chip breakpoint
The current breakpoint does not support a data value because this is not provided by the processor. Please check **Processor Architecture Manual** and reference manual of the processor.

too many data value breakpoints set
The number of desired data value on-chip breakpoint are not provided by the processor. Please check **Processor Architecture Manual** and reference manual of the processor.

address does not fit in on-chip breakpoint resource
The address or address range of the current breakpoint does not fit into the processor's breakpoint registers. Please check if the address or address range meets the alignment/range requirements of the processor (e.g. if the breakpoint consists of an address value and address mask register). See **Processor Architecture Manual** and reference manual of the processor.

address does not fit in on-chip trigger resource
The address or address range of the current breakpoint does not fit into the processor's breakpoint registers. Please check if the address or address range meets the alignment/range requirements of the processor (e.g. if the breakpoint consists of an address value and address mask register). See **Processor Architecture Manual** and reference manual of the processor.

address does not fit in on-chip breakpoint range resource
The address or address range of the current breakpoint does not fit into the processor's breakpoint registers which can handle only ranges. Please check if the address or address range meets the alignment/range requirements of the processor (e.g. if the breakpoint consists of an address value and address mask register). See **Processor Architecture Manual** and reference manual of the processor.

address range too large to fit in on-chip breakpoint range resource
The address range is too large to fit into the processor's breakpoint registers. Please check if the address or address range meets the alignment/range requirements of the processor (e.g. if the breakpoint consists of an address value and address mask register). See **Processor Architecture Manual** and reference manual of the processor.

address range does fit in bit masked on-chip breakpoint resource

The address does not fit into a bit mask. Either enable conversion of addresses or change the address to fit into a binary mask. Please check if the address or address range meets the alignment/range requirements of the processor (e.g. if the breakpoint consists of an address value and address mask register). See **Processor Architecture Manual** and reference manual of the processor.

address range does fit in bit masked on-chip trigger resource
The address does not fit into a bit mask. Either enable conversion of trigger addresses or change the address to fit into a binary mask. Please check if the address or address range meets the alignment/range requirements of the processor (e.g. if the breakpoint consists of an address value and address mask register). See **Processor Architecture Manual** and reference manual of the processor.

.address range can not be converted to fit in on-chip breakpoint resource
Address conversion must be enabled to support this type of breakpoint. See **Processor Architecture Manual** and reference manual of the processor.

.address range can not be converted to fit in on-chip trigger resource
Address conversion must be enabled to support this type of trigger breakpoint. See **Processor Architecture Manual** and reference manual of the processor.

.address range does not convert good enough to fit in on-chip breakpoint resource
The converted address would be too much outside the expected range. Please check if the address or address range meets the alignment/range requirements of the processor (e.g. if the breakpoint consists of an address value and address mask register). See **Processor Architecture Manual** and reference manual of the processor.

illegal address for on-chip breakpoint
An on-chip breakpoint was about to be set to an address which is not allowed due to limitations of the processor. Please check **Processor Architecture Manual** and reference manual of the processor.

data does not fit in bit masked on-chip breakpoint resource
The data value assigned to the on-chip breakpoint does not meet the data format provided by the on-chip breakpoint resource on the processor. The breakpoint consists of value and mask register. Either enable conversion of data values or change the data to fit into a binary mask. Please check **Processor Architecture Manual** and reference manual of the processor.

data does not fit in bit masked on-chip trigger resource
The data value assigned to the on-chip breakpoint does not meet the data format provided by the on-chip breakpoint resource on the processor. The breakpoint consists of value and mask register. Either enable conversion of data values or change the data to fit into a binary mask. Please check **Processor Architecture Manual** and reference manual of the processor.

data does not fit in ranged on-chip breakpoint resource
The data value assigned to the on-chip breakpoint does not meet the data format provided by the on-chip breakpoint resource on the processor. The breakpoint can handle data ranges. Either enable conversion of data values or change the data to fit into a range. Please check **Processor Architecture Manual** and reference manual of the processor.

data does not fit in ranged on-chip trigger resource

The data value assigned to the on-chip breakpoint does not meet the data format provided by the on-chip breakpoint resource on the processor. The breakpoint can handle data ranges. Either enable conversion of data values or change the data to fit into a range. Please check **Processor Architecture Manual** and reference manual of the processor.

signed data not supported by on-chip breakpoint resource
The data value assigned to the on-chip breakpoint does not meet the data format provided by the on-chip breakpoint resource on the processor. The breakpoint can not handle signed comparisons. Please check **Processor Architecture Manual** and reference manual of the processor.

signed data not supported by on-chip trigger resource
The data value assigned to the on-chip breakpoint does not meet the data format provided by the on-chip breakpoint resource on the processor. The breakpoint can not handle signed comparisons. Please check **Processor Architecture Manual** and reference manual of the processor.

data width not supported by on-chip breakpoint resource
The data value assigned to the on-chip breakpoint does not meet the data format provided by the on-chip breakpoint resource on the processor. Please check **Processor Architecture Manual** and reference manual of the processor.

data width not supported by on-chip trigger resource
The data value assigned to the on-chip breakpoint does not meet the data format provided by the on-chip breakpoint resource on the processor. Please check **Processor Architecture Manual** and reference manual of the processor.

NOT data is not supported by on-chip breakpoint resource
The data value assigned to the on-chip breakpoint does not meet the data format provided by the on-chip breakpoint resource on the processor. Please check **Processor Architecture Manual** and reference manual of the processor.

NOT data is not supported by on-chip trigger resource
The data value assigned to the on-chip breakpoint does not meet the data format provided by the on-chip breakpoint resource on the processor. Please check **Processor Architecture Manual** and reference manual of the processor.

ranged or masked data not supported by on-chip breakpoint resource
The data value assigned to the on-chip breakpoint does not meet the data format provided by the on-chip breakpoint resource on the processor. Please check **Processor Architecture Manual** and reference manual of the processor.

data does not fit in on-chip breakpoint resource
The data value assigned to the on-chip breakpoint does not meet the data format provided by the on-chip breakpoint resource on the processor, e.g. if the breakpoint consists of value and mask register. Please check **Processor Architecture Manual** and reference manual of the processor.

data does not fit in on-chip trigger resource
The data value assigned to the on-chip breakpoint does not meet the data format provided by the on-chip breakpoint resource on the processor, e.g. if the breakpoint consists of value and mask register. Please check **Processor Architecture Manual** and reference manual of the processor.

exclude breakpoints not available

An on-chip breakpoint was about to be set to an exclude address which is not possible on the processor. Please check **Processor Architecture Manual** and reference manual of the processor.

exclude breakpoint not possible
An on-chip breakpoint was about to be set to an exclude address which is not allowed due to limitations of the processor. Please check **Processor Architecture Manual** and reference manual of the processor.

exclude breakpoint required
This breakpoint type can only be set with /Exclude option due to limitations of the processor. Please check **Processor Architecture Manual** and reference manual of the processor.

only single exclude breakpoint of a certain type possible
Setting multiple exclude breakpoints is not possible. Please check **Processor Architecture Manual** and reference manual of the processor.

cannot match TraceON and TraceOFF breakpoints to pairs
The target requires that TraceON/TraceOFF breakpoints operate as pairs. Please check **Processor Architecture Manual** and reference manual of the processor.

no context comparator possible
The on-chip breakpoint unit does not support context comparators to set task specific breakpoints. Please check **Processor Architecture Manual** and reference manual of the processor.

no virtual machine comparator possible
The on-chip breakpoint unit does not support virtual machine comparators to set machine specific breakpoints. Please check **Processor Architecture Manual** and reference manual of the processor.

context comparator is disabled
Using the context ID register of the processor is disabled in **Break.CONFIG** window.

virtual machine comparator is disabled
Using the machine ID register of the processor is disabled in **Break.CONFIG** window.

cannot translate machine into machine ID value
A machine can not be translated to the appropriate machine ID register value.

cannot translate task into context ID value
A task can not be translated to the appropriate context ID register value.

no master comparator available
The on-chip breakpoint unit does not support bus master comparators. Please check **Processor Architecture Manual** and reference manual of the processor.

no master comparator possible
The on-chip breakpoint unit does not support bus master comparators for this breakpoint type. Please check **Processor Architecture Manual** and reference manual of the processor.

no on-chip breakpoint of this type possible
In the current system state, or dependent on the architecture, a breakpoint of the desired type can not be set. Please check **Processor Architecture Manual** for details.

on-chip breakpoint resource sharing conflict

The on-chip breakpoint about to be set uses on-chip breakpoint resources which are already used by other breakpoints. Open **Break.List** window and delete unneeded breakpoints. Please check **Processor Architecture Manual** and reference manual of the processor.

on-chip breakpoint resource *<resource>* sharing conflict
The on-chip breakpoint about to be set uses on-chip breakpoint resources which are already used by other breakpoints. Open **Break.List** window and delete unneeded breakpoints. Please check **Processor Architecture Manual** and reference manual of the processor.

Using reserved breakpoint resource *<resource>* may produce unpredictable results
The resource may be used by another function of the debugger.

selective trace conflict, cannot mix program and data trace control
Controlling the trace in this way is not really supported by the complex trigger. Please check **Processor Architecture Manual** and reference manual of the processor.

the trace produced by the complex trigger may be unreliable
Controlling the trace in this way is not really supported by the complex trigger. The resulting trace my be unpredictable. Please check **Processor Architecture Manual** and reference manual of the processor.

no more counter resources available to detect target stop by ETM
The trigger may still work, but the debugger will not be able to detect the reason for the stop when the ETM trigger stops the cores.

no more counter resources available to detect breakpoint hit by ETM
The trigger may still work, but the debugger can not detect that a certain breakpoint implemented by the ETM was hit.

mixing of program and data breakpoints may produce unpredictable results
The combination of program and data related breakpoints may not produce the expected results. Please check **Processor Architecture Manual** and reference manual of the processor.

complex trigger can not stop core
The complex trigger unit cannot stop the core. Please check **Processor Architecture Manual** and reference manual of the processor.

complex trigger has no such address comparator
The complex trigger unit has no appropriate resource. Please check **Processor Architecture Manual** and reference manual of the processor.

trace trigger and break can not be mixed
The complex trigger unit can either cause a trace trigger or a break, but not both actions at the same time. Please check **Processor Architecture Manual** and reference manual of the processor.

too many counters used
The complex trigger unit runs out of counters. Please check **Processor Architecture Manual** and reference manual of the processor.

count too large
The count is too large to fit into the counters of the complex trigger unit. Please check **Processor Architecture Manual** and reference manual of the processor.

too complex expression
The expression in a complex trigger statement is too complex to fit into the available resources. Please check **Processor Architecture Manual** and reference manual of the processor.

too complex simple trigger expression
The expression in a complex trigger statement is too complex to fit into the available resources. Please check **Processor Architecture Manual** and reference manual of the processor.

too complex simple trigger expression for core
The expression in a complex trigger statement is too complex to fit into the available resources. Please check **Processor Architecture Manual** and reference manual of the processor.

too many single-shot comparators required
Simplify the expression or use **NoSingleShot** in condition to avoid using the single-shot comparator by default. Please check **Processor Architecture Manual** and reference manual of the processor.

too many resources required
Running out of resource registers (ETMv4). Simplify the complex trigger. Please check **Processor Architecture Manual** and reference manual of the processor.

JoinCORE actions are not possible
Use SplitCORE when possible.

cross core actions not possible
Controlling the action on one core by another core is not possible. Only local actions - within a core - possible.

core clock unknown
Enter the core clock with **Trace.CLOCK** command.

time counters not possible
Core clocks may still work.

too many inputs used at resource *<resource>*
Simplify the expression. Please check **Processor Architecture Manual** and reference manual of the processor.

no such signal available
Please check **Processor Architecture Manual** and reference manual of the processor.

complex trigger state not reachable
The state of a state machine in a complex trigger can not be reached and was optimized away.

action not allowed outside sequencer states
The action must be inside a certain sequencer state.

condition not allowed outside sequencer states
The condition must be inside a certain sequencer state.

GOTO not possible
The state machine of the target does not support this state change.

too many states in state machine
The state machine of the target does not support so many states.

no more counter resources available for expression extension
Try to simplify the expression or reduce the number of used counters.

no more counter resources available for TraceON/TraceOFF control
Try to reduce the number of used counters.

no more counter resources available for SPOT detection
Try to reduce the number of used counters.

complex trigger has errors
The complex trigger had errors. Run **Break.Program** again and fix the errors or use **Break.CLEAR** to delete the complex trigger.

internal error, please contact technical support
The error should not happen, please contact the technical support.

action statement outside IF condition
Action statements must be placed after an **IF** statement that defines the condition for the actions.

no actions defined for this condition
Action statements must be places after an **IF** statement that defines the condition for the actions.

closing bracket for parameters missing
The parameters of a condition have no matching closing bracket.

opening bracket expected
The parameters of a condition need to be included in brackets. Conditions without parameters need empty brackets.

invalid name
Names for counters, flags and state machines can only include the letters 'a'...'z', '0'...'9' and '_'.They can not start with '0'...'9'.

expected memory access class only
The memory access class specifies the ZONE, e.g. S: or U:.

expected comparison operator and value
The operator and value defines the counter operation, e.g. COUNT(X>100).

same counter used in different modes
A counter is used in different modes. This is not allowed.

counter does not exist
The counter needs to be used in a condition expression to be defined.

counter type does not allow increment

Time or clock counters can only be enabled, but can not increment. Use an event counter to count events.

counter type does not allow enable
An event counter can not be enabled. Define a clock counter to count core clocks.

flag does not exist
The flag needs to be used in a condition expression to be defined.

state-machine level does not exist
The specified name does not exist as trigger level name.

state-machine level does already exist
The specified name does already exist as trigger level name.

emulation running
The current run-control command (Go/Step) can only be executed if the target program is stopped for the debugger.

emulation not running
The current run-control command (Break) can only be executed if the target program is running.

target processor in reset
The current run-control command (Go/Step/Break) can only be executed if debugger established a connection to the target (SYStem.Mode UP)

background emulation running
The command **Go.BackGround** was called, but the background task was already running.

background emulation not running
The command **Break.BackGround** was called, but the background task was already running.

cannot determine big/little core

core powered down
The selected core of a multicore SOC has no power.

core clock down
The selected core of a multicore SOC has no clock.

core security code mismatch
The user given security code doesn't match the security code of the selected core or another security mechanism is active which does not allow the debugger to access the core.

access timeout, processor running
The current command causes an access to a target resource which currently can not be accessed because the target application is running. Access to memory while the target is running is controlled by **SYStem.MemAccess**.

time-out, cannot set registers
A timeout occurred while trying to write to a register. Please check **Processor Architecture Manual** for more information or contact technical support (support@lauterbach.com).

no access to target CPU
The CPU did not grant access to the addressed resource. Please check **Processor Architecture Manual** for details.

protected memory access error
The debug hardware tried to access protected memory areas and failed. Please see **Processor Architecture Manual** for details.

debug hardware sequencer error
Internal error. Please contact technical support (support@lauterbach.com).

debugmonitor communication error
An error occurred while communicating with the monitor program. Please check **Processor Architecture Manual** for details.

target power fail
The current **SYStem.Mode** command failed, because target power was detected to be turned off. Please turn on target power. If the error persists, check if the VCC pin on the debug connector is powered (without and with debug cable connected).

target reset fail
An unexpected target reset occurred, which caused that the debugger lost control over the CPU.

target power or reset fail
The debugger lost control over the CPU, either caused by power down or target reset.

waiting for reset timed out
A reset-related timeout occurred. Please check target configuration and check **Processor Architecture Manual** for further information.

CPU clock fail
The debugger lost control over the CPU because a clock needed for debugging is disabled or not functional.

dual-port fail
The dual-port access failed. This can be either a problem with the external clock or the CPU is in a state, where the emulator cannot gain access to the databus. Check also the target-appendix for your emulation-head.

internal dual-port fail
The dual-port access failed. This can be either a problem with the external clock or the CPU is in a state, where the emulator cannot gain access to the databus. Check also the target-appendix for your emulation-head.

dual-port busy
The dual-port access failed. This can be either a problem with the external clock or the CPU is in a state, where the emulator cannot gain access to the databus. Check also the target-appendix for your emulation-head. Only a warning message.

target system locked
The SYStem.LOCK command was called before the current command. All commands that access the target are locked. Execute **SYStem.LOCK OFF.**

triggered before start, reinit trigger
ICE/FIRE: The trigger has to be reset before the Go command can be executed. Please see **Processor Architecture Manual** for details.

out of emulation memory
ICE/FIRE: It was attempted to map more emulation memory than available.
PowerDebug: The debug interface ran out of memory. Please contact technical support (support@lauterbach.com).

out of flag memory
ICE/FIRE: It was attempted to use more flag memory than available.

out of breakpoint memory
ICE/FIRE: It was attempted to set more breakpoints than possible by hardware.

software breakpoints not possible with current system setting
The current system settings/state does not allow software breakpoints. Please refer the corresponding **Processor Architecture Manual** for more details about restrictions for breakpoints.

break not allowed
The current command is not allowed while **SYStem.CpuBreak Denied** is set.

operation would cause spot - which is not allowed
The current command is only allowed when **SYStem.CpuSpot Enable** is set.

operation would cause single spot - which is not allowed
The current command is only allowed when **SYStem.CpuSpot Enable** or **SYStem.CpuSpot Single** is set.

operation would cause target called spot - which is not allowed
The current command is only allowed when **SYStem.CpuSpot Denied** is not set.

operation requires target memory access while core is running
The current command is only allowed when an appropriate memory access mode is set.

target stopped for stop and resume operation - which is not allowed
The resume operation is not possible with the **SYStem.CpuSpot Denied** setting.

operation not allowed when CTS is active
The current command is not allowed when the context tracking system is enabled. Execute **CTS.OFF** and try again.

operation not allowed when the virtual execution mode is active
The current command is not allowed when the virtual execution mode is enabled. Execute **VE.OFF** and try again.

double clock fail
The external and internal clocks failed. Please contact technical support (support@lauterbach.com).

debug hardware configuration error
Please check **Processor Architecture Manual** for details.

Power Debug: The selected CPU does not match the detected CPU.

ICE/FIRE: Please check if the DIP-switches are set correctly.

emulation monitor cycle fail
The target cpu makes no cycles during execution of the emulation-monitor. Problem can be caused by special states of the CPU (idle/powerdown modes).

emulation monitor break error
The emulation control monitor didn't get control after a breakpoint. Problem can be caused by special states of the CPU (idle/powerdown modes), bad CPU-clock or a general failure of the probe.

emulation monitor entered by reset
The emulation/rom monitor was re-entered because a target reset occurred.

emulation monitor entered by interrupt/trap
An interrupt or trap caused the emulation/rom monitor to be entered.

emulation memory refresh fail
The refresh of the dynamic internal emulation memory failed. In most times this is caused by too long cycles in the target. The problem can be solved either by supplying the correct signal from the target to shorten the cycles (wait/dtack) or by shortening the 'timeout'-value in the SYStem-menu.

emulation system memory fail
The system memory of the ECU unit failed. Please contact technical support (support@lauterbach.com).

emulation monitor fail, executed trap
The internal emulator monitor has executed a trap. Please contact technical support (support@lauterbach.com).

debug port fail
The debug-port of the emulation CPU failed. Check processor, socket and target-clock.

debug port error #<*number*>
The debug-port of the emulation CPU failed. Check processor, socket and target-clock.

debug port locked
The debug-port of the emulation has been locked (security active).

debug port problem
The debug-port of the emulation CPU failed. Check processor, socket and target-clock.

target reset detected
An active reset signal was detected.

debug port failed after reset
Check BDM connection.

debug port RTCK failed
Check RTCK connection or chip settings.

target multicore configuration wrong
Check multicore or chip settings.

debug port time-out at <*address*>
The debug port of the emulation CPU had time-out during memory read (no DSACK or BERR signal generated). Check chip-select and bus monitor function. Normally the FRZBM bit in the SIM module should be disabled (set to No).

subcore communication time-out
The communication to the main core debugger timed out.

emulation fail, strobes always active
The strobes (AS,DS,R/W) are always active.

target idle
The core is in idle mode. The action is not possible in this mode.

debug monitor fail
The emulation monitor is not working correctly. If system is working with internal clock, then there is a problem with the clock from the target. If the system also fails in stand-alone mode, there is either a hardware problem in the emulator or the CPU is defective. Check also the target-appendix for your emulation-pod.

cannot access fpu
The access to a floating point register failed. Please check **Processor Architecture Manual** for details.

command is not allowed in current system access mode
The command is only allowed if dual-port access mode is set to denied.

dual port access not allowed
The dual-port access is currently locked and can not be used. Please check **Processor Architecture Manual** for details.

FPU off
The FPU-commands are switched off, use **FPU.ON** to activate the FPU.

target program was not running till now
Because of a processor restriction, the required operation (e.g. FPU access) is only possible after the processor was started at least once with the GO command.

verify error at address <*address*>
The data was not correctly written into the memory.

code at software breakpoint has changed at address <*address*>
The previously written software breakpoint code was overwritten by the target program. This is just a warning.

attempt to set breakpoint inside slot instruction
The breakpoint is not allowed here. Move the breakpoint to a different location.

no memory mapped at address <*address*>
Tried to load data by dual-port access at a location, where no memory was mapped.

memory contents not valid
The address about to be accessed is currently invalid. Please check if the address is correct and if the memory at this address is already initialized.

virtual memory contents (VM:) not valid
It was attempted to access an address in virtual memory, which was not yet initialized with data.

memory access not allowed
Tried to access memory that is mapped as volatile or denyacces map MAP commands.

variable not existing
The variable could not be found in the current context. In some cases variables are not available because of compiler optimizations.

cannot single step this instruction
The instruction at the current Program Pointer is illegal or can not be single stepped due to debug limitations. Use the Data.List window and Mode.Mixed command to display the assembly code that shall be executed.

too many return points in this function
The **Go.Return** command could not be executed because too many return points were found.

no return points in this function
The **Go.Return** command could not be executed because no return points were found.

already at return point of this function
The **Go.Return** command could not be executed because the program is already at a return location.

cannot determine return points for function *<function>*
The **Go.Return** command could not be executed because the return points could not be determined.

not in function
The **Go.Return** command could not be executed because a previous function entry could not be determined.

no more reachable targets from this code
The **Step.Diverge** command could not be executed because all parts that can be reached by the currently active stepped code have already been reached.

no such stack frame
Commands using information from stack (e.g. Go.UP, Register.UP, Register.Down) could not be executed because the stack is invalid.

already at entry point of this function
The **Go.BackEntry** command could not be executed because tthe program is already at the entry location.

memory segment invalid or limit exceeded (segmentation error)
A protected mode address cannot be translated to a linear address because either the given segment descriptor is not applicable for the address or the address offset exceeds the limit specified by the segment . A descriptor being not applicable can be due to a invalid segment, the wrong type of the segment. Check the segment registers using the command **MMU.view**. If a descriptor table walk is triggered by the translation, check the global and/or local descriptor table using the commands **MMU.DUMP.GDT** or **MMU.DUMP.LDT**.

MMU translation failed
The logical address could not be translated into a valid physical address.

Space ID invalid
The space ID given is not known to the system. Check the currently valid space IDs using the command **TASK.List.tasks**.

machine not defined
The MMU format for the specified guest machine ID or the intermediate page table (second translation stage) for this machine ID is undefined. Please ensure that you specify the MMU setup with **MMU.FORMAT** for all virtual machines (guests) used.

Impossible to configure FPGA because debugger is attached
The FPGA can not be loaded because there is an active debug connection to the FPGA. Please execute **SYStem.Down** first.

data alignment error
Some CPU's can write word or long-word data only aligned to even addresses.

target vector table not valid
Some processors need specific entries in the vector table before executing real-time commands.

write line on chip not enabled
Please enable the write enable line of the processor before doing write accesses.

Emulator synch timeout
Several TRACE32 instances are configured for synchronized Go/Step or Break (see **SYNCH** window) and one if the connected instances does not respond. Please check if all connected TRACE32 instances are running and if the firewall is configured to allows opening a UDP socket.

Emulator usr access disabled
The target program that does the USR memory access returned with an error. Because of this error USR access has been disabled. Or the USR memory access is not configured.

bus error generated by CPU
A memory access caused a bus timeout. If unsure which memory access caused the problem, please check the **SYStem.LOG.List** window.

bus error received by monitor
A memory access performed by the emulation/rom monitor caused a bus error.

address error received by monitor
An interrupt occurred while the monitor tried to read/write memory. Check **SYStem.LOG.List** window to see which addresses were accessed.

illegal instruction error received by monitor
An illegal instruction interrupt occurred while the monitor code was executed. The monitor program is probably corrupted, or the current core (machine) state does not allow some instructions used by the monitor.

error response from monitor
The monitor returned with an error. The source of this error could not be determined.

syntax error in floating-point number
The floating point number in the command line contains a syntax error.

unknown register name: *<regname>*
The register name passed as argument of the function Register is unknown.

invalid bus width
The bus width specified in the command or function is invalid or missing.

debug port secured
The debug port of the processor is secured. Please see **Processor Architecture Manual** on how to unsecure the debug port of the processor.

debug function blocked by device security
The debug port of the processor, or a part of the debug port is secured. Please see **Processor Architecture Manual** on how to unsecure the debug port of the processor.

invalid MMU translation for this CPU
The address translation specified with **TRANSlation.Create** is not possible on this processor. Please check the address ranges.

syntax error in symbol
The debug symbol entered with the current command contains a syntax error.

symbol is not a function
The symbol specified in the current command is not a function name. Check spelling.

symbol is not a modul
The symbol specified in the current command is not a module name. Check spelling.

no source information
There is no source information loaded for that module. Loading sources is only possible if sourcelines were already loaded.

ambiguous symbol
There are multiple symbols addressed by this name. Use the sYmbol.Browser to display these symbols. Use the sYmbol.MATCH command to change the symbol selection strategy.

ambiguous method name, need prototype/class
The method name is ambiguous (e.g. operator overloading). Select the desired method through the **sYmbol.List** window.

line not found
The line number specified with the current command is invalid.

symbol is line number
The current command expected a symbol name, but a line number was entered.

symbol not found
The symbol name in the current command is unknown. Check spelling.

symbol not found in this context
The symbol specified in the current command or function is unknown in the current context.

symboltype not displayable

variable has no address
The variable about to be displayed is currently hold in a register. Therefore no address is available.

symbol has no address or address range
Some symbols have no assigned address range. This can have several reasons e.g. loaded symbol file doesn't contains address information about the actual used symbol.

```
e.g.    PRINT sYmbol.RANGE(\\programname)
```

no access to that symbol

no such language loaded
Loaded languages are displayed in the module table. The name is always a code the language and the used compiler, e.g. MCC68K, CCC68K, UBROF-C.

local symbol file <*filename*> not found
The symbol file with the specified name could not be found. Please make sure that the file is available and the path names are correct.

cannot change from short to long line number format
Without **Puzzled** or **COLumns** option a short linenumber format is used. If more than one file is to be loaded, all file must have either short or long format.

function and scope information cannot be aligned
The information about functions and scopes (lexical blocks) cannot be matched.

too many types in one module for packing
The command **Data.LOAD** was called with option `/PACK` and the file contained too many types. Please load the file without option `/PACK`.

wrong character for bit mask (only '0', '1' or 'X' allowed)
The argument of the current command expects a bit mask. Only '0', '1' and 'x'/'X' (don't care) are allowed.

wrong accessmode
The memory class used for the current access is not supported by the eprom simulator.

invalid operation code in bank file
The first byte of the banking definition file is must be the bank mode. The bank file is probably corrupted.

DTACK error reported by CPU
While executing a memory access, the data acknowledge signal was not received. Use **MAP.DenyAccess** to prevent accessing unimplemented memory addresses. It is recommended to implement the board hardware in a way that a bus timeout is generated when unimplemented memory addresses are accessed.

Unknown bitstream
Not supported bitstream format. Please verify that you use a valid Xilinx bitstream file.

Used SMP cores/threads not defined
The error occurs if command **SYStem.CONFIG.CoreNumber** was entered and the system mode changed with missing information about which hardware cores/threads are used for this SMP debug session. Use commands **CORE.NUMber** or **CORE.ASSIGN** to define the used hardware threads/cores.

Core inactive
There is no access to the core due to missing power, missing clock, the core is held in reset or the core sleeps.

Failed to write *<value>* to target register *<register_/name>*
For some reason the debugger was not able to write the specified value to a target register.
If you debugging a customized SoC please check with your SoC designers. Otherwise contact technical support (support@lauterbach.com).

address range incorrectly aligned or length inconsistent with access width
Address range borders must match the intended alignment (short, word, quad addresses).

```
e.g. wrong:   Data.Test P:0x101--0x1fe /Long
     correct: Data.Test P:0x100--0x1ff /Long
```

Options may only be specified once
Each option can only specified once. Also, some options preclude other options: /KEY and /NOKEY are exclusive, as are /USER, /USERL and /USERH.

key mismatch
The key provided by the user does not match the expected one.

For /USERH, bits 7..0 of the specified user code must be zero
For /USERL, bits 31..8 of the specified user code must be zero
Specify a 32-bit value for each of these options, with the bits that are not within the correct bit range set to zero.

unknown symbol filter
The **symbol filter** with the specified name does not exist. Create the **symbol filter** before using it.

fatal error #<*code*>, contact technical support (support@lauterbach.com)
Internal error. Please contact technical support (support@lauterbach.com).

no debugging device attached
TRACE32 could not connect to the debug interface or simulator or API. Please check the startup configuration (usually config.t32). If TRACE32 was started in simulator mode, this error occurs if the installed version of TRACE32 does not provide an instruction set simulator for the selected architecture. Please check www.lauterbach.com if an instruction set simulator is offered for this architecture and if an update is available for download.

too many different compilers
Several debug symbol files were loaded, which were compiled with too many different compiler versions. If possible, use the same compiler version or load less debug symbol files.

division by zero
The mathematical operation of the current PRACTICE command caused a division by zero.

too many source paths
The maximum number of allowed source paths was exceeded. Reduce the number of source paths.

PODBUS sharing violation
Internal error. Please contact technical support (support@lauterbach.com).

fatal PLD load failure
Possible causes are:

- Podbus configuration error. Please check if (optional) `USE=` entry or Trace32Start configuration is correct (PowerDebug and PowerDebug USB1 modules need two digits for `USE` entry)
- Podbus connector problem. Please check if all cables are attached properly.
- Problem with power supply. Please check if the appropriate power supply is used.
- Hardware failure. Please contact technical support (support@lauterbach.com).

thermal overload, check cooling system
Please check if the debug hardware is placed so that is does not overheat, heat sinks have sufficient airing and the cooling fan (if available) is spinning. In the latter case, please contact technical support (support@lauterbach.com).

fatal error in SUCO startup, contact technical support (support@lauterbach.com).
A fatal error occurred while starting a sub controller. Please check Podbus connections, other cable connections, power supply, connection to host and configuration file (config.t32). If the problem persists, please contact technical support (support@lauterbach.com).

fatal error (<*code*>) in SUCO startup, contact technical support (support@lauterbach.com)
A fatal error occurred while starting a sub controller. Please check Podbus connections, other cable connections, power supply, connection to host and configuration file (config.t32). If the problem persists, please contact technical support (support@lauterbach.com).

ICD Trace hardware (PLD) problem
An internal error occurred. Please contact technical support (support@lauterbach.com).

ICD Trace hardware (RAM) problem
An internal error occurred. Please contact technical support (support@lauterbach.com).

Sub-Controller RAM error, contact technical support (support@lauterbach.com)
An internal error occurred. Please contact technical support (support@lauterbach.com).

fatal error in SubController startup (RAM content verify), contact technical support (support@lauterbach.com)
An internal error occurred. Please contact technical support (support@lauterbach.com).

passive mode

passive probe not connected

active probe not connected

function not implemented
The requested operation is not implemented or not possible with the debug hardware.

function not available on this device
The requested operation is not supported by the target device.

function disabled due to license problems
TRACE32 detected a corrupt debug cable. As a result several fuctions have been disabled. Please check if your debug cable is plugged correctly and if it is original equipment from Lauterbach. Please contact technical support (support@lauterbach.com).

command not implemented in sub-processor
The command to be executed is not implemented in the current software version. Please contact technical support (support@lauterbach.com).

sub-processor stack overflow
Internal error. Please contact technical support (support@lauterbach.com).

register set not defined
Internal error. Please contact technical support (support@lauterbach.com).

Software problem in mapper, please contact technical support (support@lauterbach.com)
Internal error. Please contact technical support (support@lauterbach.com).

decompression failed
Internal error. Please contact technical support (support@lauterbach.com).

fatal memory overflow
Internal error. Please contact technical support (support@lauterbach.com).

out of local memory
Internal error. Please contact technical support (support@lauterbach.com).

Event counter error, contact technical support (support@lauterbach.com)
Internal error. Please contact technical support (support@lauterbach.com).

self test of eprom simulator failed
The selftest detected a hardware error of the eprom simulator. Please contact technical support (support@lauterbach.com).

fatal software configuration error, contact technical support (support@lauterbach.com)
Internal error. Please contact technical support (support@lauterbach.com).

fatal error in port analyzer
An internal error occurred. Please contact technical support (support@lauterbach.com).

name information table overflow, contact technical support (support@lauterbach.com)
Internal error. Please contact technical support (support@lauterbach.com).

function information table overflow, contact technical support (support@lauterbach.com)
Internal error. Please contact technical support (support@lauterbach.com).

fatal error in symbol table
Internal error. Please contact technical support (support@lauterbach.com).

type information table overflow, contact technical support (support@lauterbach.com)
Internal error. Please contact technical support (support@lauterbach.com).

inconsistent TRACE32 installation, please re-install
Host software and driver software version don't match. This can happen if files from two installations are mixed.

only access modes <access_class> expected
The allowed access modes are CPU dependend.

failed to init counter system
Internal error. Please contact technical support (support@lauterbach.com).

unknown debug signal
You have chosen a name of a debug signal, which is unknown to TRACE32. Please see command **JTAG.PIN** for a list of known signals.

analyzer is armed, no access possible
While the trace (Analyzer, ART, Logger, etc) is armed, accesses to the trace buffer are not allowed. Please wait until the trace stops, or turn trace off manually using **Trace.Off**.

analyzer file not loaded, use Trace.FILE command
A Trace.* (Analyzer, ART, Logger, etc) command was called with option /FILE, but no trace file was loaded using **Trace.FILE** yet.

analyzer not readable
Internal error, please contact technical support (support@lauterbach.com).

profiler already in use with different setup
The profiler can only operate on one counter with one sample-rate.

unknown counter name or wrong mode
The name must be the same that is used in the analyzer programming sequence. The mode of the counter must also by the same.

unknown flag name
The name must be the same which is used in the analyzer programming (CTU) sequence.

unknown level name
The specified level name of the current command does not exist. Check spelling.

incompatible analyzer data
A Trace (Analyzer, ART, Logger, etc) recording was about to be loaded, but the file to be loaded is not compatible to the selected Trace method.

analyzer counter overflow
The counters of the analyzer are 48 bit wide, while the arithmetic of the system is 32 bits. This error indicates that the accessed counter value cannot be represented by a 32 bit value.

corresponding analyzer list window required
To search the items **TIme.FUNC** or **TIme.MARK**, a Analyzer.List window displaying the searched items is necessary.

too many items
The maximum number of items selected for the Trace.List window was exceeded.

WARNING: analyzer timestamp overflow, times may be incorrect
The timestamp unit of the analyzer detected a timestamp overflow. Please be aware that the times displayed in the analyzer listing can be wrong. If this error occurs, use shorter trace times.

fatal error in hypertrace search (<*address*>), contact technical support (support@lauterbach.com)
Internal error. Please contact technical support (support@lauterbach.com).

fatal error in hypertrace (<*address*>,<*hrecord*>,<*record*>), contact technical support (support@lauterbach.com)
Internal error. Please contact technical support (support@lauterbach.com).

fatal error in analyzer, stackmode overrun, contact technical support (support@lauterbach.com)
Internal error. Please contact technical support (support@lauterbach.com).

no such channel
The channel name specified as argument of the current command does not exist. Check spelling.

Trigger delay out of range
The specified trigger delay is out of the allowed range.

bookmark not found
The bookmark with the specified name was not found. Open the **BookMark.List** window for a list of bookmarks.

target clock frequency unknown
Timestamps for the trace cannot be calculated because the target core clock is not known.The clock can be defined with the **Trace.CLOCK** command.

emulation extension chip (EEC) not available
No Emulation Extension Chip (EEC) was found. Probably the selected CPU is not an Emulation Device.

record number is out of range
The record number passed to this command or PRACTICE function does not exist. Either no trace recording was made at all, or the record number is outside the available record number range.

MCDS internal error - contact technical support (support@lauterbach.com)
A software-internal error has occurred within the MCDS-related part of TRACE32 PowerView. Contact technical support (support@lauterbach.com) and provide detailed information on your TRACE32 system, configuration and the entered commands or scripts.

MCDS feature not supported
This MCDS feature is not supported by the TRACE32 PowerView software. Contact technical support (support@lauterbach.com) for more information on whether this feature is supported by a newer software version.

MCDS feature not available
The requested MCDS resource or programming is not available. Try to find another solution.

MCDS resources already in use
One or more of the MCDS resources required for this programming are already used for another purpose. Try to find an alternative solution or programming.

error enabling MCDS
Enabling MCDS failed, check hardware and configuration. Check the debug connection, your device, the CPU selection and your hardware design.

error writing MCDS configuration to target
The device reported an error while writing the MCDS configuration. Check the debug connection, the device and the selected CPU. Note that the MCDS setup may be cached internally so this message may not appear directly when configuring.

error reading MCDS configuration from target
The device reported an error while reading the MCDS configuration. Check the debug connection, the device and the selected CPU.

MCDS internal error, failed to initialize resource management
TRACE32 PowerView failed to initialize the resource management. Contact technical support (support@lauterbach.com) and provide detailed information on your TRACE32 system, configuration and the entered commands or scripts.

MCDS resource already in use
The requested MCDS resource is already in use, try to use another configuration.

Trace test failed: Don't know where to execute the test code
If the command Trace.AutoFocus or Trace.TestFocus is executed, TRACE32 attempts for load a test program to the target RAM. TRACE32 looks for RAM at the memory addressed by the PC, the stack pointer or the *<address_range>* entered with the command. If this error message appears TRACE32 is unable to load the test program. Please make sure that PC and/or stack pointer are pointing to RAM or specify a valid *<address_range>*.

Trace test failed: not enough samples in the trace
Repeat the command Trace.AutoFocus/Trace.TestFocus. The persistence of this problem indicates that the clock or enable signals might have a problem. Check the following:

- trace port enabled?
- buffer devices enabled?
- trace clock enabled?
- threshold out of signal range?

When using the Preprocessor for ARM-ETM with AUTOFOCUS the ETM clock frequency can be double-checked with AutoFocus diagnosis tool (~~/demo/etc/diagnosis/autofocus/afdiagnosis.cmm). If the clock frequency measured by the diagnosis tool is nonsense most likely the clock signal is causing the problem. If the diagnosis tool measures the expected frequency, the enable signals might be the problem. For example for an ETMv.3 architecture the TRACECTRL signal (= Trace Signal 2) is expected on pin 36 of the trace port connector (please refer to **'Technical Data'** in **"ARM-ETM Trace"** (trace_arm_etm.pdf). Some general guidelines on how to proceed are provided under '**Diagnosis**' in **"ARM-ETM Trace"** (trace_arm_etm.pdf).

Trace test failed: trace empty
See **Trace test failed: not enough samples in the trace.**

Trace test failed: trace empty, probably no trace clock
See **Trace test failed: not enough samples in the trace.**

Trace test failed: trace control pin not working
The trace control pin (TRACECTL for ETMv3 or PSx for ETMv1) is not working.

Trace test failed: there is no test program available for the selected core
In order to start the self calibration (Trace.AutoFocus) or to start the data eye finder (Trace.TestFocus) TRACE32 requires a test program running on the target. For the CPU you selected via SYStem.CPU there is no test program available. Please contact the technical support (support@lauterbach.com).

Trace test failed: test not available
Software implementation issue. Please contact technical support (support@lauterbach.com).

Trace test failed: pin connection error
This message will occur, if one or more trace channels are stuck to either VCC or GND or are not changing independently. This might be caused by:

- unsupported ETM mode
- trace port not enabled
- wrong connector pinout
- shorts between data pins TP[..]

Repeat the command Trace.AutoFocus/Trace.TestFocus for both Trace.TERM ON and OFF as well as at the lowest target frequency possible. The persistence of this problem indicates that trace channels might be floating or that there is coupling between trace channels. Some general guidelines on how to proceed are provided under **'Diagnosis'** in **"ARM-ETM Trace"** (trace_arm_etm.pdf).

Trace test failed: can't get program flow trace
The TRACE32 software is unable to complete the program flow calculation. Please contact technical support (support@lauterbach.com).

Trace test failed: trace contains no program flow
This problem can either be the trace port configuration or the trace port sampling.

Trace test failed: trace too short to verify the correctness of the program flow
This problem is likely to be frequency dependant. Repeat the command Trace.AutoFocus/Trace.TestFocus at the lowest target frequency possible. If the problem persists repeat it for both Trace.TERM ON and OFF. Some general guidelines on how to proceed are provided under **'Diagnosis'** in **"ARM-ETM Trace"** (trace_arm_etm.pdf).

Trace test failed: some errors detected in the program flow
Check the following:

- trace port enabled
- timing violations on status pins PS[..], TS
- threshold out of signal range

Trace test failed: many errors detected in the program flow
See **Trace test failed: some errors detected in the program flow.**

Trace test failed: no data flow information in the trace
Check the following:

- trace port enabled
- timing violations on data pins TP[..]
- threshold out of signal range

See **Trace test failed: no data flow information in the trace.**

Trace test failed: FIFOFULL and wrong test pattern
The trace port is configured too slow or too narrow to trace the required information.

Trace test failed: no data patterns
See **Trace test failed: no data flow information in the trace.**

Trace test failed: not enough data patterns
See **Trace test failed: no data flow information in the trace**.

Trace test failed: errors in the data flow
See **Trace test failed: no data flow information in the trace.**

Trace test failed: wrong test pattern
Data cache not updated -> disable data cache for testing. Also see **Trace test failed: no data flow information in the trace.**

Trace test failed: wrong test pattern sometimes on all pins
See **Trace test failed: wrong test pattern.**

Trace test failed: wrong test pattern sometimes on some pins
See **Trace test failed: wrong test pattern.**

Trace test failed: test pattern not sufficient
This message occurs, if one or more trace channels are stuck to either VCC or GND or are not changing independently. Opposed to **Trace test failed: pin connection error** this just means that the test pattern generated by the test program were not sufficient to test all trace channels. In other words the trace channels that could be tested were tested successfully, but not all channels could be tested. For example this is the case for ETMv.1 in WideDemux mode, when the instruction cache is off. If supported by the target enable the instruction cache to avoid this message.

Trace test failed: test code not running properly.
If the command Trace.AutoFocus or Trace.TestFocus is executed, TRACE32 attempts to load a test program to the target RAM and execute it. The test program implements an endless loop that generates a worst case test pattern on the trace port. This message means that the test program finished prematurely. Use the /keep option to analyze the cause of the problem. See **Trace.AutoFocus** for details.

Trace test failed.
One or more of the above messages have occurred after executing Trace.AutoFocus or Trace.TestFocus. Use the command **AREA.view** to display all error messages

Trace test failed: can´t read data eye border
When using the Preprocessor for ARM-ETM without AUTOFOCUS the Trace.AutoFocus command will attempt to set up the best possible reference voltage (the only parameter that can be changed for that preprocessor). This message indicates that the algorithm was unable to find a reference voltage. Repeat the **Trace.AutoFocus** command and contact technical support (support@lauterbach.com), if the issue persists.

Emulator shadow RAM not accessed

Emulator shadow RAM busy

Emulator shadow RAM not mapped

Mapping not possible
The desired mapping can not be programmed, e.g. because of address range restrictions or because the target is running. Please see **Processor Architecture Manual** for details.

WARNING: refresh setup inconsistent, turned off
ICE: The DRAM refresh configuration was inconsistent and therefore was turned off. Please see **Processor Architecture Manual** for details.

Cannot map writeflag when readflag is already mapped
Before mapping writeflag, remove the readflag mapping.

Cannot map readflag when writeflag is already mapped
Before mapping readflag, remove the writeflag mapping.

Cannot map emulation memory in BreakOnly access mode
Mapping emulation memory is not allowed in this mode. Check **Processor Architecture Manual** for details.

Cannot enable BreakOnly mode when emulation memory is mapped
Before enabling BreakOnly mode, unmap emulation memory first.

function <*function*> not patched
The specified function could not be patched, e.g. because of access restrictions.

tag patch area overflow
The maximum amount of possible patches was exceeded.

function tagging not implemented for this processor
Function tagging is not implemented for this processor. Please contact technical support (support@lauterbach.com) regarding availability of updates.

don't know how to change instruction at <*address*>
Function tagging failed, because the instruction at the specified address could not be changed. Please contact technical support (support@lauterbach.com).

no emulation memory mapped at this address
The fast data exchange function requires emulation memory at the exchange location.

emulation monitor not responding
The communication to the emulation monitor could not be established. Please check **Processor Architecture Manual** for details.

emulation memory not correctly mapped
The emulation memory has not been mapped correctly. Please check **Processor Architecture Manual** for details.

WARNING: relation TIMEOUT to TIMEREQ is critical
The 'timereq' value must be larger than the 'timeout' value. If the value is smaller and the probe in used in 'Request'-mode the dual-port fail error will occur before the timeout is generated. If the target hardware has a working timeout-circuit, which terminates the bus-cycle (maybe with a buserror) this will make no problems.

no line selected

not triggerd

not breaked

turbo download memory violation
The turbo download failed due to a address collision between data and turbo download agent. Check and change your SYStem.Option.MonBase setting.

turbo download failed
The turbo download failed due to multiple handshake errors between the turbo download agent on the target and the debugger. Try to reduce the JTAG frequency or disable the turbo option for this target.

Error Messages

no function prototype to supply overlaid arguments to function

arguments don't match function prototype

address of call dummy routine not defined, use SETUP.VARCALL

return value of function cannot be identified

no global variable <*name*>

base class <*type*> not inherited

not pointer to method

negative sized type encountered

target function call not possible with this command

Use the commands **Var.set** or **Var.Call** for function calls instead.

processor running after target function call

target function call failed, PC not at expected location

invalid parameter for target function call

result too wide

The size of the result for this command is limited to 128 bytes.

don't know vptr of <*type*>

failed to trace back over bad stackframe

The access to a variable, burried in the stack frame cannot be made.

no stack frame for variable <*name*> found

static member <*name*> of class not found

method <*name*> of class not found

function <*name*> not found

no type <*typename*>

no enumeration member <*name*>

no virtual function found

memory allocation in target failed

illegal cast to <*type*>

<*type*> is not an array

<*type*> is not a pointer to member

trying to dereference a non pointer

expression has no value

expression has no address

trying to access element <*name*> of non structure expression

trying to access element <*name*> of non structure pointer expression

no field <*name*> in structure

no member or method <*name*> in class

<*name*> is not a static member of the class

incompatible pointer types

trying to call a non function

array index out of range

trying numeric operation with non numeric arguments

invalid operator combination

<*type*> is not a class

<*member*> is not a valid destructor

invalid character constant

invalid character in symbol

invalid digit in numeric expression

<*name*> is not a valid boolean constant (TRUE or FALSE)

cannot assign set to a non set variable

syntax error in expression

empty expression

extra characters after expression

no language for expressions selected

Use the command **sYmbol.LANGUAGE** to select a language for parsing expressions.

macro expansion failed

out of memory during expression evaluation

feature not implemented

fatal error in expression parser, contact technical support (support@lauterbach.com)

Error Messages

subcommand expected

option already set

One and the same option could be selected only one time.

```
e.g.   b.l   /C /R /c
                   ^ this option is set twice
```

only one option usable

One option mutual excludes other options.

address too big

The address value exceeds the highest CPU address.

wait number too big

Wait cycle number exceeds the valid range of 0..250.
If only 4K block waits are permitted then the valid range is 0..15.

wait number expected

whole number expected

Only whole numbers are accepted.

accessmode <access_class> not allowed

address, address range or accessmode expected

at address <*address*> no free RAM left

At these address no free mapperram of the pretended type left (e.g. static data ram or dynamic break ram).
This error message can also appear at the commands "set fixed breakpoint (**Break.Set** <*address*>)", "set temporary breakpoint (**Break.** <*address*>)" or "go temporary breakpoint (**Go** <*address*>)", because these commands automatically map breakram if necessary.
If the command **MAP.state** shows still free ram, this breakram can be used. Than the mapper has to be switched into the slow mode with **MAP.Mode SLOW**. Only in the slow mode it is possible to map every 4K-block to any logical 4k-block address.

no static flagram at address <*address*>

No static flag ram exists at these already mapped address.

Actually at the logical address <*address*> a dynamic ram is mapped (break, data ram or both). Mixed ram types aren't possible.

static flagram at address <*address*> already used

The mapping system uses (locked) the static flag ram caused from an earlier command "**MAP.Flag** *<address>*".

A solution is the complete unmapping and the renewed mapping of the 4K-block including the address.

static breakram at address <*address*> already used

The mapping system uses (locked) the static break ram caused from an earlier command "**MAP.Break** *<address> [\Static]*".

A solution is the complete unmapping and the renewed mapping of the 4K-block including the address.

dynamic breakram at address <*address*> already used

The mapping system uses (locked) the dynamic break ram caused from an earlier command "**MAP.Break** *<address> [\Dynamic]*".

A solution is the complete unmapping and the renewed mapping of the 4K-block including the address.

all breakrams at address <*address*> already used

The mapping system uses (locked) all break rams caused from earlier commands.A solution is the complete unmapping and the renewed mapping of the 4K-block including the address.

at address <*address*> already dynamic breakram mapped

The static break ram could only be mapped on the logical address <*address*> if previously the dynamic ram was unmapped with the command "**MAP.NoBreak beginaddress4kblock--endaddress4kblock**".

Perhaps the command option "*\Static*" is wrong.

breakram at address <*address*> must be DYNAMIC

The break and data ram must have the same ram type (Static or Dynamic).

The data ram was already mapped as a dynamic ram and therefore the break ram must be also a dynamic ram.

The command option "*\Static*" is wrong.

at address <*address*> already static breakram mapped

The dynamic break ram could only be mapped on the logical address <*address*> if previously the static ram was unmapped with the command "**MAP.NoBreak beginaddress4kblock--endaddress4kblock**".

Perhaps the command option "*\Dynamic*" is wrong.

breakram at address <*address*> must be STATIC

The break and data ram must have the same ram type (Static or Dynamic).
The data ram was already mapped as a static ram and therefore the break ram must be also a static ram.
The command option "*Dynamic*" is wrong.

dynamic breakram at address <*address*> already used

The mapping system uses (locked) the dynamic break ram caused from an earlier command "**MAP.Break** *<address> [\Dynamic]*".
A solution is the complete unmapping and the renewed mapping of the 4K-block including the address.

static dataram at address <*address*> already used

The mapping system uses (locked) the static data ram caused from an earlier command "**MAP.Data** *<address> [\STATIC]*".
A solution is the complete unmapping and the renewed mapping of the 4K-block including the address.

dynamic dataram at address <*address*> already used

The mapping system uses (locked) the dynamic data ram caused from an earlier command "**MAP.Data** *<address> [\Dynamic]*".
A solution is the complete unmapping and the renewed mapping of the 4K-block including the address.

all datarams at address <*address*> already used

The mapping system uses (locked) all data rams caused from earlier commands. A solution is the complete unmapping and the renewed mapping of the 4K-block including the address.

at address <*address*> already dynamic dataram mapped

The static data ram could only be mapped on the logical address *<address>* if previously the dynamic ram was unmapped with the command "**MAP.NoData beginaddress4kblock--endaddress4kblock**".
Perhaps the command option "*Static*" is wrong.

dataram at address <*address*> must be DYNAMIC

The flag, break and data ram must have the same ram type (Static or Dynamic). The flag respectively the break ram was already mapped as a dynamic ram and therefore the data ram must be also a dynamic ram.
The command option "*Static*" is wrong.

at address <*address*> already static dataram mapped

The dynamic data ram could only be mapped on the logical address *<address>* if previously the static ram was unmapped with the command "**MAP.NoData beginaddress4kblock--endaddress4kblock**".
Perhaps the command option "*Dynamic*" is wrong.

dataram at address <*address*> must be STATIC

The flag, break and data ram must have the same ram type (Static or Dynamic). The flag respectively break ram was already mapped as a static ram and therefore the data ram must be also a static ram.

The command option "*\Dynamic*" is wrong.

dynamic dataram at address <*address*> already used

The mapping system uses (locked) the dynamic data ram caused from an earlier command "**MAP.Data** *<address> [\Dynamic]*".
A solution is the complete unmapping and the renewed mapping of the 4K-block including the address.

unmapping of dataram is not allowed while cpu is running

no dataram is mapped at address <*address*>, although it is necessary

no suitable finemapper is free at address <*address*>

internal error : EMU_MAP_531

internal error : EMU_MAP_532

too many SPLIT commands please start MAP.RESET

Too many SPLIT commands. Please start **MAP.RESet** and begin again with the SPLIT commands.
This is only a solution if split commands which are not necessary are omitted (e.g. the same address area is splitted multiple (repeatedly)).
`E.g. "MAP.SPlit d:1000--2fff"  and  "MAP.SPlit sd:1000--2fff"`
It is better to use the commands "MAP.SPlit sd:1000--2fff" and "MAP.SPlit ud:1000--2fff"

command not allowed while CPU is running

combined accessmodes are not allowed, e.g. "d:" or "S:"

no 4K-block beginaddress

no 4K-block endaddress (range end address)

range to big - mapper maximum address is exceeded (while mirroring)

source and destination area are overlapping

not enough memory for the command

too many temporary breakpoints (maximum 10)

no flagram is mapped at address <*address*>, but it is necessary

For setting flags it is necessary to have flag ram mapped on the desired addresses.
This can be achieved by :

a) the additional command option "**/M**" (automatical flag ram mapping), i.e.
the command **FLAG.Set** itself maps flag ram at addresses on which no flag ram exists

or

b) the command "**MAP.Flag** <*address*>". It maps flag ram for the whole 4K-block, which includes the address.

command not allowed while CPU is running

The command is not allowed while the CPU is running.

The flag system has to be switched on or off only at stopped program. Otherwise you cause undefined states while triggering at read respectively write flags.

the option <*option*> excludes this option

All writtens option build together a "print condition".

If the flags from an address fulfill these "print condition", then a corresponding line in the list window is generated. Of course it is not possible to generate a condition like "flag set" and at the same time "flag not set" (e.g. /READ and /NOREAD).

only finemapper waitcycle number <*waits*> usable

For all single addresses (bytewise switch on of waitcycles) respectively address range with waitcycles > 15 there exists only o n e finemapper waitcycle number, i.e. it is not possible to configure at 2 single addresses 2 different numbers of wait cycles.

no ram exist

No ram exists or no ram of the desired type exists (only for **MAP.DEFAULT**).

at address <*address*> waitnumber not configurable

At the address the number of waitcycles isn't configurable.

The effective (actually) number of waitcycles is the highest number of the "bytewise waits" and the "4k-block waits".

The current number of 4k-block waits is higher than the current bitwise waits. Therefore the finemapper waits (bytewise waitcycles) must be higher than the previously set waitcycles belonging to the 4k-block.

only accessmodes C: or E: are allowed

Only accessmodes C: or E: are allowed, because the command affects always all accessmodes (at maximum 4) at the same time.

e.g. 80386 version: As default there are 2 splitted (separated) physical address spaces P: and IO:. The workbench definition affects always P: and IO:. In this case a "map.pre 100000++0fffff" is wrong because the default accessmode is P:.

no 1MB-blockbegin- or endaddress

The address range border isn't MB-blockbegin respectively endaddress.

The interval borders of the actual address range are corresponding to workbench begin respectively end addresses, i.e. interval borders must be 1MB-begin respectively endaddresses. E.g. 100000--2fffff means that 2 workbenches should be created at the logical addresses 100000--1fffff and 200000--2fffff.

only 16 workbenches possible

In the whole logical address range of the CPU only 16 1MB-address areas can be selected these ram could be mapped later.

only 16 addressranges are allowed at these command

at address <*address*> no workbench is defined

If the desired address isn't in a defined workbench then no ram can be mapped respectively the attributes INTERN, PROTECT and NOACCESS can't be switched on. Just so flags or breakpoints can't be set because without workbench definition no flag respectively break ram can be mapped.

Likewise no workbench can be delete at an area where no workbench was defined before.

<*address*> is no MB-block borderaddress

If no workbench is defined for a 1MB-block then waits can be set only uniform for the whole 1MB-block.

accessmode C: or E: expected at address <*address*>

If no workbench is defined for a 1MB-block and the desired waitcycle number > 15 then the waits can only be set for all accessmodes at the same time (in this case is only C: respectively E: permitted).

<*address*> is no MB-block borderaddress or no MB range

If no workbench is defined for a MB-block then breakpoints can be changed (set or deleted) only uniform in the whole MB-block.

at address <*address*> no workbench is defined

If the desired address isn't inside a defined workbench it is not possible to create a temporary breakpoint.

static flagram at address <*address*> already used

The mapping system uses (locked) the static flag ram caused from an earlier command "**MAP.Flag** *<address> [\Static]*".

A solution is the complete unmapping and the renewed mapping of the 4K-block including the address.

dynamic flagram at address <*address*> already used

The mapping system uses (locked) the dynamic flag ram caused from an earlier command "**MAP.Flag** *<address> [\Dynamic]*".

A solution is the complete unmapping and the renewed mapping of the 4K-block including the address.

at address <*address*> already dynamic flagram mapped

The static flag ram could only be mapped on the logical address *<address>* if previously the dynamic ram was unmapped with the command "**MAP.NoFlag beginaddress4kblock--endaddress4kblock**".

Perhaps the command option "*\Static*" is wrong.

flagram at address <*address*> must be DYNAMIC

The flag, break and data ram must have the same ram type (Static or Dynamic).The break respectively the data ram was already mapped as a dynamic ram and therefore the flag ram must be also a dynamic ram.
The command option "*Static*" is wrong.

at address <*address*> already static flagram mapped

The dynamic flag ram could only be mapped on the logical address <*address*> if previously the static ram was unmapped with the command "**MAP.NoFlag beginaddress4kblock--endaddress4kblock**".
Perhaps the command option "*Dynamic*" is wrong.

flagram at address <*address*> must be STATIC

The flag, break and data ram must have the same ram type (Static or Dynamic). The break respectively the data ram was already mapped as a static ram and therefore the flag ram must be also a static ram.
The command option "*Dynamic*" is wrong.

at address <*address*> no workbench is defined

The range which should be mirrored must be totally inside a defined workbench.

at address <*address*> no workbench is defined

The destination addressrange of the mirroring must be completely inside a defined workbench.

option already set

Option was already set or is included in another set option
(for example. "/DEFAULT /WAIT" or "/DATA /ALL").

command Go.Return not possible while CPU is running

command Go.Next not possible while CPU is running

internal error : <text>

internal runtime error ! Please contact the manufacturer !

no 4K-block beginaddress

Bank areas must begin at 4K-blockbeginaddresses.

no 4K-block endaddress (range end address)

Bank areas must end at 4K-blockendaddresses.

range beginaddress too big - maximumaddress is exceeded

Range begin address is higher than the maximum CPU address.
Bank areas must be inside the logical CPU address area.

range endaddress too big - maximumaddress is exceeded

Range end address is higher than the maximum CPU address.
Bank areas must be inside the logical CPU address area.

bank-range too small

Bank area is too small. Bank areas must be bigger than 4KB.

bank-range too big

Bank area is too big. Bank areas must not bigger than 16KB.

address <address> too big

The address value exceeds the biggest address for bit or byte breakpoints.

bit/byte breakpoint command not allowed at the moment (because of SLOW MODE)

internal error : EMU_MAP_577

Only data breakpoints for bit and byte breakpoints usable.

internal error : EMU_MAP_578

No breakram can be mapped on the accessmodes bit and byte.

command not possible - mirroring only in the standard workbenches permitted

Mirroring is only permitted in the standard workbenches (16 workbenches from address 0 to 0ffffff).

no dyn. RAM mapable, because SYStem.Access is Denied

<address >is no 4K-block beginaddress

Areas must begin at 4K-blockbeginaddresses.
Only 4k-wise setting or reseting of attribute BUS8, BUS16, BUS32, BUSEXT, ONCHIPP, DMA, OPFETCH, 4K blockwaits, ... is available.

<address> is no 4K-block endaddress (range end address)

Areas must end at 4K-blockendaddresses.
Only 4k-wise setting or reseting of attribute BUS8, BUS16, BUS32, BUSEXT, ONCHIPP, DMA, OPFETCH, 4K blockwaits, ... is available.

memory access not possible, dual port access is locked

The command is locked at the moment because it uses the dualport access and the dualport access is blocked at the moment. The dualport lock is active if e.g. **SYStem.Access** is Denied and the CPU is running.

no type for temporare breakpoint given

address value <value> bigger than <n> bits

The address value exceeds the biggest address for operand breakpoints.

e.g. b.s 10000 /od

bitbreakpoints are currently locked

The bitbreakpoints are only usable in the bondout version.

```
e.g.    b.s  b:10.1 /ob        or
        b.d  b:                or
        b.d  c:    /obx
```

this breakpoint combination not allowed

Normal breakpoint types like /Program couldn't be combined with special breakpoint types like /OD or /OAW. Please use instead 2 breakpoint set commands.

```
e.g.    b.s 100 /P /od          wrong
        b.s 100 /p              ok
        b.s 100 /od
```

this breakpoint combination not allowed

Several special breakpoint types like /OD or /OAW aren't usable at the same time. Please use instead 2 breakpoint set commands.

```
e.g.    b.s 100 /oaw /od        wrong
        b.s 100 /oaw            ok
        b.s 100 /od
```

<address> is no 1MB-block beginaddress

Areas must begin at 1MB-blockbeginaddresses.

Outside of workbenches only 1MB-wise setting or reseting of attribute BUS8, BUS16, BUS32, BUSEXT, ONCHIPP, DMA, CACHE, BURST, ACK, ... is available.

<address> is no 1MB-block endaddress (range end address)

Areas must end at 1MB-blockendaddresses.

Outside of workbenches only 1MB-wise setting or reseting of attribute BUS8, BUS16, BUS32, BUSEXT, ONCHIPP, DMA, CACHE, BURST, ACK, ... is available.

<*address*> is no MB-block borderaddress

The address range border isn't MB-blockbegin respectively endaddress.

If no workbench is defined for a 1MB-block then the memory splitting can be done only uniform for the whole 1MB-block.

command is currently not possible because system access is denied

The command is locked at the moment because it uses the dualport access and the dualport access will be blocked if the CPU will run.

logical address expected

The command expects only logical addresses.

```
E.g. p:0--1000 instead of ap:0--1000
```

this breakpoint combination not allowed

Fast breakpoint types like /FP1 couldn't be combined with fast breakpoint types like /FR2. Please use instead 2 breakpoint set commands.

```
e.g.   b.sfast 100 /FP1 /fr2          wrong
       b.sfast 100 /fp1               ok
       b.sfast 100 /fr2
```

maximum number of configuration items exceeded

The maximum number of 10 configuration item was exceeded.

```
e.g. flag.listfunction %b r w rnw nrw %p r w rnw nrw %b rnw %p rnw %b r
```

no more workbench numbers expected

Actually more workbench numbers given than necessary.

```
eg.  map.pre c:0--1fffff 1 2 3
                           ^ error
```

more workbench numbers expected

Actually less workbench numbers given than necessary.

```
e.g.  map.pre c:0--1fffff 1
                         ^ error
```

only 16 workbench numbers are allowed at these command

workbench number <n> already used

The workbench number was already used in a command before.

Workbench numbers couldn't be used twice.

```
e.g.  map.pre c:0--1fffff 1 2
      map.pre c:400000++0fffff 1
                           ^ error
```

for 1MB block workbench number <n> already defined

For the given address range a workbench number was already defined in a command before.

Address ranges couldn't have more than one workbench.

```
e.g.  map.pre c:0--1fffff 1 2
      map.pre c:0++0fffff 3
                         ^ error
```

command locked - no shadow RAM available

Shadow command group is locked, because no corresponding hardware was detected.

address range exceeds blockmaximum (<maxblock>)

Currently it isn't possible to define addressrange over several blocks at once. Please use instead 2 defines.

mapmode FAST locked due to a special mapper configuration (MAP.MIrror)

Currently it isn't possible to switch the mapper into FASTmode. This locking was caused from a MAP.MIrror command which needs to have the mapper in SLOWmode.

```
E.g.  MAP.MIrror P:0--0fff P:1000       ; only allowed in slowmode
```

this special mapper configuration not possible in mappermode FAST

The given MAP.MIrror command needs to have the mapper in SLOWmode. The current mappermode is switched to FASTmode.

```
E.g.  MAP.MIrror P:0--0fff P:1000       ; only allowed in slowmode
```

this special mapper configuration not possible with DRAM

The given MAP.MIrror command needs to have a DRAM with smaller blocksize or a SRAM modul.

```
E.g.  MAP.MIrror P:0--0fff P:1000       ; only allowed with SRAM
```

warning: this special mapper configuration not possible with SRAM in FASTMODE

The given MAP.MIrror command will work in the correct intension if either the mappermode will be switched to SLOW mode (MAP.Mode Slow) or only the DRAM will be used for all RAM mapping commands (option /Dynamic).

```
E.g.  MAP.MIrror P:0--0fff P:10000       ; only allowed with SRAM
```

warning: this special mapper configuration not possible with DRAM

The given MAP.MIrror command will work in the correct intension if only the SRAM will be used for all RAM mapping commands (option /Static).

```
E.g.  MAP.MIrror P:0--0fff P:10000       ; only allowed with SRAM
```

command currently locked

The given MAP.DEFault or MAP.Intern command is locked due to the special activated mapper mode. In this mode no emulation memory is permitted to be switched internal. It could be used only as a kind of shadow memory.

error during fulldualported memory access

An unexpected error occured during accessing a fulldualported memory.

break memory access not possible, dualport access is locked

Some of the given addresses need dualport access to a special break memory which is currently locked or fulldualported breakRAM must be mapped which is impossible in the current state.

The command is aborted because it has to use the dualport access and the dualport access is blocked at the moment. The dualport lock is activ if e.g. **SYStem.Access** is Denied and the CPU is running.

flag memory access not possible, dualport access is locked

Some of the given addresses need dualport access to a special flag memory which is currently locked or fulldualported flagRAM must be mapped which is impossible in the current state.

The command is aborted because it has to use the dualport access and the dualport access is blocked at the moment. The dualport lock is active if e.g. **SYStem.Access** is Denied and the CPU is running.

normal and bondout breakpointtypes are mixed

internal error: EMU_MAP_613

The normal and bondout (e.g. OAR) aren't served simultaneously.

bondout breakpointtypes locked due to nonbondoutCPU

internal error: EMU_MAP_614

The bondout (e.g. OAR) could be used only with BondoutCPU.

bitmask unexpected

internal error: EMU_MAP_615

Unexpected resulttype bitmask is used for function call.

only addresses and addressranges allowed

address value <value> bigger than <n> bits

internal error: EMU_MAP_616

The address value exceeds the biggest address for operand breakpoints.

```
e.g.  addr od P:10000
```

OBR or OBW breakpoints used together with other breaktypes

internal error: EMU_MAP_617

The OBR/OBW breakpoints must be set separately from other breakpoint types.

bitmask only permitted with OD breaktypes

internal error: EMU_MAP_618

Bitmask expressions could be used only in combination with OD* breakpoints (e.g. ODL).

wrong workbench number (0x00..0x0f)

The given actually physical workbenchnumber is wrong. Only values inside the range 0x00..0x0f are allowed.

```
e.g.  map.pre c:0x0--0x1fffff 0x1f 0x0a
                              ^ error
```

up number is too big

The given actually value is too big. Only 32bit values are allowed.

```
e.g.  go.up 123456789012345
            ^ error
```

step number is too big

The given actually value is too big. Only 32bit values are allowed.

```
e.g.  step 123456789012345
            ^ error
```

internal error : EMU_MAP_ERROR

Error Messages

internal error : <*text*>

Please contact technical support (support@lauterbach.com).

assembler program too long

The generated assembler code exceeds the internal code buffer size.

too many ORG commands

The number of ORG commands used inside the present assembler program are exceeding the limit.

syntax error

no more input

operand required

',' expected

invalid addressing mode

no valid instruction for actual selected CPU type

CPU type is not implemented or activated for this command or syntax

no value 0 permitted

bitnumber between 0 and 7 expected

only values between 0 and 8 expected

the maximum of the value bit-number is 4 bit (0..0xf)

the maximum of the value is 1 bit (0/1)

the maximum of the value is 2 bit (0..0x3)

the maximum of the value is 3 bit (0..0x7)

the maximum of the value is 4 bit (0..0xf)

5 bit value expected (0..0x1f)

5 bit value expected (1..0x1f)

the maximum of the value is 6 bit (0..0x3F)

the maximum of the value is 7 bit (0..0x7F)

the maximum of the value is 8 bit (0..0xff)

the maximum of the value is 9 bit (0..0x1ff)

the maximum of the value is 10 bit (0..0x3FF)

the maximum of the value is 12 bit (0..0xFFF)

the maximum of the value is 16 bit (0..0xFFFF)

the maximum of the value is 18 bit (0..0x3FFFF)

the maximum of the value is 20 bit (0..0xFFFFF)

the maximum of the value is 22 bit (0..0x3FFFFF)

the maximum of the value is 24 bit (0..0xFFFFFF)

the maximum of the value is 32 bit (0..0xFFFFFFFF)

32-bit register not allowed, is only for protected mode

2-bit register expected

16-bit register expected

32-bit register expected

16-bit address too far away, only a relative-8-bit-operand is allowed

32-bit address not allowed in this mode or for this command

address displacement too hugh in this mode or for this command

address too long

address illegal, even number is necessary

address illegal, is too far away from the current addr position

Ax or PC required

Ax required

Ax, Dx or '[' required

Ax, Dx or ']' required

[Offset : Width] required

constant too large, not allowed in this mode or for this command

constant is too large or too small

32-bit constant not allowed in this mode or for this command

constant required

the type or name of the Co-processor register is wrong

only values (1,2,-1,-2) expected

Dx required

':' required

the page number x is wrong

the next must be the DP register and colon (:)

Dx or Ax required

floating point register required

index or register for index not exist (Reg32_Bit*2, *4 or *8 allowed)

index register is used twice

index not exist

index not exist (*2, *4 or *8 allowed)

keyword combination not allowed

keyword combination not exist

keyword not exist

Dx, Ax or const required

const, Ax, Dx or PC required

const, Ax, Dx, '[' or PC required

const negativ

const, Ax, Dx or '[' required

scale factor expected

mnemo or label required

Ax only if the FPIAR is the single register selected

operand ACC not permitted

"MOV A,ACC" isn't a valid instruction

no [offset : width]

no static constant

no bitaddressable direct address

'(' required

the next must be an offset and colon (:)

Offset must be divisible by 4

only 32-bit register, register combination not allowed

Operand combination not allowed, not exist

operand expected

operand data format required

the type or name of the register is wrong

register direct not allowed for one operand

the type or name of the register is wrong

register PR expected

wrong register

wrong register name or type in register list

only values 0..3 for register expected

register R6 expected

The addressing with displacement is only possible with register R6.

the type or name of the register is wrong for the command or operand

register name FPSCR expected

register name FPUL expected

this register name is not allowed

register name XMTRX expected

register combination not allowed

same register

Dx, Ax or PC required

register not allowed for one operand

R0 or R1 expected

For addressing mode "@Rr" register R0 or R1 expected

the relative value is larger than 8 bit, it's too large or too small for short

REPEAT xxx TIMES is wrong (xxx=MRW or #data5Bit)

REPEAT register TIMES is wrong (register=MRW)

REPEAT xxx TIMES is wrong (xxx=#data5Bit)

')' required

')' or ',' equired

']'or ',' required

shift register command need another syntax (LSL, LSR, ASR, ROR)

Dx only if a single FPcr (FPCR, FPSR, FPIAR) is selected

bitvalue is too large for this command

'*', ')', ',', ':' or ']' required

'*', ']' or ',' required

'-', ',' or ']' required

'.', '*', ':', ',', ')' or ']' required

'.', '*', ',' or ')' required

':', ']', ')' or ',' required

invalid operand data format

invalid registersequence

error in scale
scale 68000 : '*1', scale 68020 : '*1', '*2', '*4', '*8'

invalid name on position of usr0 or usr1

trap number out of valid range (0x30..0x39 and 0x40..0xff)

'W' or 'L' required

']' required

condition code AL is not permitted

register PC and SP aren't permitted

For addressing mode only registers R0 till R12 expected.
Registers R15 (PC) and register R13 (SP) aren't permitted

only 2 or 4 register for command expected

Only register pairs or double pairs expected.
e.g. {R0,R1} or {R4,R7}
Odd register names like {R1,R2} respectively hugher register lists like {R2,R7} aren't permitted.

only 4 register for command expected

Only register double pair expected.
e.g. {R2,R5} or {R4,R7}
Odd register names like {R1,R4} respectively smaller or hugher register lists like {R2,R7} aren't permitted.

only register LR expected

only register PC expected

only register SP expected

only registers MOL till M3H expected

only register R15 expected

Error Messages

internal error :

Please contact the manufacturer !

number in this context not allowed

A number is not allowed outside a name or a declaration expression.

":" expected

unexpected EOL - closing '**:**' for logical operator expected
normally ':' respectively ':' written in the next line

```
typical error:  1. line:  "s.e,  c.i intno   if  ab:A"
                2. line:  ":program"
```

unexpected character

unexpected EOF

"|" expected

unexpected character or EOL - second '|' for logical OR operator expected
normally '|' is forgotten respectively '|' is written in the next line

```
typical error:  1. line:  "s.e,  c.i intno   if  ab|"
                2. line:  "|program"
```

'^' expected

unexpected character or EOL - second '^' for logical XOR operator expected
normally '^' is forgotten respectively '^' is written in the next line

```
typical error:  1. line:  "s.e,  c.i intno   if  ab^"
                2. line:  "^program"
```

'&' expected

unexpected character or EOL - second '**&**' for logical AND operator expected
normally '&' is forgotten respectively '&' is written in the next line

```
typical error:  1. line:  "s.e,  c.i intno   if  ab&"
                2. line:  "&program"
```

'/' expected

unexpected character or EOL - second '/' character for comment begin expected
normally '/' is forgotten respectively '/' is written in the next line

```
typical error:  1. line:  "s.e,  c.i intno   if  /"
                2. line:  "/ enable sampling and increment counter intno"
```

oldfashioned operator locked in current radix mode

In the current parser mode is the old syntax of operators and operands locked. Please switch mode to CLASSIC (**SETUP.RADIX CLASSIC**) or use new syntax.

old syntax operators: N: :A: :X: :O:
new syntax operators: ! && ^^ ||

```
e.g. sample.enable  if  N:write
                        ^  error
     sample.enable  if  !write
                        ^  ok
```

Please refer for details to chapter **parser changes** too !

not enough memory (for name table)

Not enough system memory for the expanded name table available.

too many names

Too many names used inside a trigger unit progam.

keyword <*name*> isn't allowed

This keyword is locked with the current analyzer hardware.

keyword <*name*> isn't allowed

This keyword is locked with the current ICE hardware.
This address eventname can be used only in a configuration with bondout chip.

EOF expected

unrecognized symbol;
superfluous (needless) symbols at a position where EOF is expected.

declaration must stand before the first instruction or level name

At this position in the state analyzer program there is no declaration permitted. Declarations must stand before the first global instruction respectively the first level name (label).

```
e.g.  "timecounter  delay 10.ms"
      "s.e  if delay"
      "eventcounter nr_int"
```

keywords as levelnames not allowed

A reserved keyword is used as a levelname. This is prohibited.

```
e.g.   "START:  continue                  if  ab"
       "L0:     sample.enable             if  !delaycnt"
        ^
or
       "mark:   counter.increment intcnt  if  cb"
        ^
```

command expected

EOL expected

keyword for declaration required

declaration type is locked

This declaration isn't permitted with the current analyzer hardware.

data events locked with the current analyzer hardware

The data event isn't available with the current fireanalyzer hardware. This feature is released in a later hardware version.

declaration type is locked with current CPU type

The data event VDATA isn't available with the current hardware. This feature is only released in Motorola 68332 version.

separating BLANK expected

unexpected EOL - condition for command list expected

unexpected EOL - condition for command list expected
normally condition forgotten respectively condition written in the next line

```
typical error:  1. line:  "s.e,  c.i intno   if  "
                2. line:  "ab&&program"
```

unexpected EOL - command for command list expected

unexpected EOL - command for command list expected

normally command forgotten respectively command written in the next line

```
typical error:  1. line:  "s.e,  "
                2. line:  "c.i intno   if  ab&&program"
```

SAMPLE.ENABLE or COUNTER.INCREMENT only without condition

Due to hardware restrictions it isn't possible to use in the actual level the command SAMPLE.ENABLE respectively COUNTER.INCREMENT DELAY depending on a condition (HAC32).

In construction 1 only conditioned SAMPLE.ENABLE in level 0 and 1 allowed.In construction 2 only conditioned SAMPLE.ENABLE in level 0 allowed.

The command COUNTER.INCREMENT DELAY must be used always without any condition.

The command COUNTER.INCREMENT with a countername different from DELAY could be used in the levels 0 (construction 1 and 2) and in level 1 (only construction 2).

label expected

separating BLANK expected

Frequently the name of the destination level is forgotten respectively a wrong symbol is written.

```
typical error :  "goto , flag.true interrupt_occurred"
                       ^
```

keyword "IF" expected

separating BLANK expected

Frequently a wrong symbol is written instead of the condition.

```
typical error :  "goto level0 if ,"
                                 ^
```

unexpected EOL - level name for goto expected

unexpected EOL - destination level name for goto command expected
normally name forgotten respectively name written in the next line

```
typical error:  1. line:  "goto  "
                2. line:  "start"
```

unexpected EOL - condition expected

unexpected EOL - condition for command expected
normally condition forgotten respectively condition written in the next line

```
typical error:  1. line:  "goto start if "
                2. line:  "write&&ab"
```

unexpected GOTO

Due to hardware restrictions it isn't possible to use in the current level the command GOTO (HAC32). GOTO could be used only in the highest level and the destination level must be the lowest level.

Please use instead the command CONTINUE to reach the next following level.

```
e.g.    level0:  sample.enable if ab
                 goto level1   if ab
                 ^                      error
        level1:  sample.enable if bb
                 goto level0   if bb
solution:
                 continue      if ab
```

keyword "IF" expected

separating BLANK expected

Frequently a wrong symbol is written instead of the condition.

```
typical error :  "continue if ,"
                                ^
```

keyword "IF" expected

separating BLANK expected

Frequently a wrong symbol is written instead of the condition.

```
typical error :  "break if ,"
                             ^
```

":" expected (for labelend) or unknown command

Often a not existing command is used inside the trigger program.

```
E.g.    PRINT  IF  Write
        ^                error position
```

operator :O: isn't allowed

This keyword is locked with the current analyzer hardware.
Only conditions without logical OR operator are allowed.

```
e.g. sample.enable  if  write:O:read
                              ^          error
```

Please use instead 2 separate commands

```
e.g. sample.enable  if  write
     sample.enable  if  read
```

operator :XOR: isn't allowed

This keyword is locked with the current analyzer hardware.

")" expected

name of input event expected

input event not allowed

Input event isn't permitted with the current analyzer hardware configuration.

unexpected EOL - ")" expected

unexpected EOL - ")" at the end of an condition is expected
normally ")" forgotten respectively ")" written in the next line

```
typical error:  1. line:  "goto start if (int&&ab&&write"
                2. line:  ")"
```

repeated parenthesis "(" isn't allowed

Multiple parenthesis is locked with the current analyzer hardware (HAC32).

```
e.g.   sample.enable  if  !((write&&one)&&event1)
                            ^   error
       sample.enable  if  !(write&&(one&&event1))
                                   ^   error
```

only simple {! N:} allowed

This construction is locked with the current analyzer hardware.

```
e.g.  sample.enable if !!ab
                       ^       error
```

Use instead

```
      sample.enable if     ab
```

To negate the trigger condition please use a construction like

```
      sample.enable if !(!ab&&write)
```

unexpected EOL - name of input event expected

unexpected EOL- name of input event at the end of an condition is expectednormally name forgotten respectively name written in the next line

```
typically error:   1. line:   "goto start if (delay&&"
                   2. line:   "BUSA)"
```

EOL expected (no more input events)

This construction is locked with the current analyzer hardware (HAC32).

After parenthesis there is no additional condition parts permitted.

```
e.g.  sample.enable  if !(ab&&write)&&event1
                                    ^       error
```

Only constructions like

```
      sample.enable  if  !(ab&&write&&event1)
```

are permitted.

"N:" expected

The construction "(...)" is locked with the current analyzer hardware (HAC32). Parenthesis could be used only in combination with logical not {! N:}.

```
e.g.  sample.enable  if  (ab&&write)
                         ^               error
```

Only constructions like

```
      sample.enable  if  !(ab&&write)
```

are permitted.

"!(tp-condition)" only in construction 1 allowed

Due to hardware restrictions it isn't possible to use in construction 2 a condition like ! (TP) (only HAC32). This condition could be used only in construction 1.

```
e.g.  sample.enable  if !(ab&&:write)
                        ^       error in combination with construction 2
```

command not allowed

This command is locked with the current analyzer hardware or isn't permitted in the current line in combination with the other used commands.

command not allowed

unexpected EOL - subcommand expected

invalid subcommand

unexpected EOL - subcommand expected

invalid subcommand

unexpected EOL - counter subcommand expected

unexpected EOL - A subcommand of the counter command is expected.
Normally name is forgotten respectively name is written in the next line.

```
typical error :  1. line:  "counter."
                 2. line:  "increment interrupt_events if INT1"
```

invalid subcommand

subcommand is omitted or given subcommand is not correct

```
e.g.:   "c. CNT1 IF INT", "c.onn CNT1 IF INT" or "c.o n CNT1 IF INT"
```

unexpected EOL - username for counter event expected

unexpected EOL - An of the user given name for the counter event (TIMECOUNTER, EVENTCOUNTER) is expected.
Normally name is forgotten respectively name is written in the next line.

```
typical error :  1. line:  "counter.increment  "
                 2. line:  "interrupt_events   "
```

separating BLANK expected

name expected

unexpected EOL - "." expected

"." expected

unexpected EOL - subcommand for flag command expected

unexpected EOL - An subcommand for the flag command is expected.
Normally subcommand is forgotten respectively subcommand is written in the next line.

```
typical error :  1. line:  "flag.   "
                 2. line:  "true interrupt_occurred    "
```

invalid subcommand

unexpected EOL - username for flag expected

unexpected EOL - An of the user given name for the flag event is expected.
Normally name is forgotten respectively name is written in the next line.

```
typical error :  1. line:  "flag.true   "
                 2. line:  "interrupt_occurred    "
```

separating BLANK expected

Frequently the flagname is forgotten respectively a wrong symbol is written.

```
typical error :  "flag.true , interrupt_occurred"
```

name expected

unexpected EOL - level name for goto expected

unexpected EOL - destination level name for goto command expected
normally name forgotten respectively name written in the next line

```
typical error:  1. line:  "goto   "
                2. line:  "start"
```

separating BLANK expected

Frequently the name of the destination level is forgotten respectively a wrong symbol is written.

```
typical error :  "goto , flag.true interrupt_occurred"
                       ^
```

label expected

unexpected EOL - "." expected

"." expected

unexpected EOL - subcommand expected

invalid subcommand

unexpected EOL - subcommand for sample command expected

unexpected EOL - An subcommand for the sample command is expected.
Normally subcommand is forgotten respectively subcommand is written in the next line.

```
typical error :  1. line:  "sample."
                 2. line:  "enable if cnt1"
```

invalid subcommand

subcommand is omitted or given subcommand is not correct
```
e.g.:   "sample.  IF CNT1", "s.eanble IF CNT1" or "s. enable IF CNT1"
```

unexpected EOL - "." expected

"." expected

unexpected EOL - subcommand expected

invalid subcommand

name of address event expected

address or addressrange expected

The value of the defined address event is expected. Often only the access mode is left out.
```
e.g. "address ab 100--200||300--400"    instead of
     "address ab p:100--200||300--400"
```

separating BLANK expected

Frequently a wrong symbol is written instead of the address event.
```
typical error :  "address alphab , sp:1000"
                                 ^
```

separating BLANK expected

Frequently a wrong symbol is written instead of the address event name.
```
typical error :  "address , sp:1000"
                         ^
```

unexpected EOL - address event name for address event expected

unexpected EOL - address event name for address event expected
Normally name forgotten respectively name written in the next line.
```
typical error:  1. line:  "address  "
                2. line:  "ab  up:1000--5500  "
```

unexpected EOL - address or addressrange expression expected

unexpected EOL - The value of the defined address event is expected.
Normally value forgotten respectively the value stands in the next line.
```
typical error:  1. line:  "address  alphabreak "
                2. line:  "  up:1000--5500     "
```

address expression expected

The next value of the defined address event is expected behind the written ",".
Frequently the expression is from a wrong type, it's forgotten or the expression stands in the next line respectively a unexpected symbol stands in before the expression.

```
typical error :  "address  ab  P:20 , 0xff"   or
                 "address  ab  P:20,     "    or
typical error :  1. line:  "address ab P:20,"
                 2. line:  " D:0xff"
```

only one addressvalue expected

Only one addressvalue could be assigned to the special bondout address events.

```
e.g. "address OD  0x1234 0x5678"
```

instead of

```
     "address OD  0x1234"
     "address ODX 0x5678"
```

"." expected

name of channel (A or B) expected

name expected

separating BLANK expected

Frequently a wrong symbol is written instead of the trigger event name.

```
typical error :  "trig.a ,extern_started 0xff"
                          ^
```

boolean expression expected

unexpected EOL - "." expected

unexpected EOL - Normally "." is forgotten respectively it stands in the next line.

```
typical error :  1. line:  "trig  "
                   2. line:  ".a extern_started 0xff  "
```

unexpected EOL - name of channel (A or B) expected

unexpected EOL - Normally the channel name is forgotten respectively it stands in the next line.

```
typical error :  1. line:  "trig.  "
                 2. line:  "a extern_started 0xff  "
```

unexpected EOL - trigger event name for trigger event or blank expected

unexpected EOL - Trigger event name for trigger event or blank is expected
Normally the name is forgotten respectively it stands in the next line.

```
typical error :  1. line:  "trig.a  "
                 2. line:  "extern_started 0xff  "
```

reserved name used

The use of names reserved from the system as names for trigger event is forbidden.

```
E.g. "trig.a  TRUE  0xff"
```

unexpected EOL - data expression expected

unexpected EOL - The value of the defined trigger event is expected.
Frequently the expression is forgotten respectively stands in the next line.

```
typical error :  1. line:  "trig.a  extern_started"
                 2. line:  " 0xff"
```

name of channel B is locked

The channel is forbidden with the current analzer hardware configuration.

separating BLANK expected

Frequently the value of the trigger event is omitted or a wrong symbol is written instead of the trigger event.

```
typical error :  "trig.a extern_started  ,0xff"
                                         ^
```

data expression expected

The value of the defined trigger event is expected. Frequently the expression is from a wrong type or it's forgotten respectively a unexpected symbol stands in before the expression.

```
typical error :  "trig.a  extern_started  p:0xff"    or
                 "trig.a  extern_started  ,12"
```

data expression expected

The next value of the defined trigger event is expected behind the written ",". Frequently the expression is from a wrong type, it's forgotten or the expression stands in the next line respectively a unexpected symbol stands in before the expression.

```
typical error :  "trig.a  extern_started  20 , p:0xff"   or
                 "trig.a  extern_started  20, "          or
typical error :  1. line:  "trig.a  extern_started  20,"
                 2. line:  " 0ff"
```

"." expected

separating BLANK expected

name expected

The name for the data event is expected.

valid DATA extension expected (sometimes CPU mode dependent)

The parsed DATA extension isn't allowed generally or is in the current CPU mode not possible.
e.g. 68302: If **SYStem.Line BusSize 8** than the data extensions B0, B1, W0 aren't allowed any more.

unexpected EOL - data expression expected

unexpected EOL - The value of the defined data event is expected. Frequently the expression is forgotten respectively stands in the next line.

```
typical error :  1. line:  "data.b  upper_char"
                 2. line:  " 'A'--'Z'"
```

unexpected EOL - "." expected

unexpected EOL - Normally "." is forgotten respectively it stands in the next line.

```
typical error :  1. line:  "data  "
                 2. line:  ".b upper_char 'A'--'Z'"
```

unexpected EOL - DATA extension expected

unexpected EOL - Normally the DATA extension is forgotten respectively it stands in the next line.

```
typical error :  1. line:  "data.  "
                 2. line:  "b upper_char 'A'--'Z'"
```

unexpected EOL - data event name for data event or blank expected

unexpected EOL - Data event name for the data event or blank is expected.
Normally the name is forgotten respectively it stands in the next line.

```
typical error :  1. line:  "data.b  "
                 2. line:  " upper_char 'A'--'Z'"
```

reserved name used

The use of names reserved from the system as names for data event is forbidden.

```
E.g. "data.b  TRUE  1"
```

separating BLANK expected

data expression expected

The value of the defined data event is expected. Frequently the expression is from a wrong type or it's forgotten respectively a unexpected symbol stands in before the expression.

```
typical error :  "data.b  extern_started  p:0xff"    or
                 "data.b  extern_started  ,12"
```

data expression expected

The next value of the defined data event is expected behind the written ",". Frequently the expression is from a wrong type, it's forgotten or the expression stands in the next line respectively a unexpected symbol stands in before the expression.

```
typical error :  "data.b  extern_started  0x20 , p:0xff"   or
                 "data.b  extern_started  0x20, "          or
typical error :  1. line:  "data.b  extern_started  0x20,"
                 2. line:  " 0xff"
```

unexpected EOL - username for HWME event expected

unexpected EOL - An of the user given name for HWME event is expected.
Normally name is forgotten respectively name is written in the next line.

```
typical error:  1. line:  "HWME  "
                2. line:  "NMI  0x0800  "
```

separating BLANK expected

reserved name used

The use of names reserved from the system as names for HWME event is forbidden.

```
E.g. "hwme  TRUE  0x0800"
```

name expected

The name of the HWME event is expected.

unexpected EOL - value for HWME event expected

unexpected EOL - The value of the defined HWME event is expected. Frequently the expression is forgotten respectively stands in the next line.

```
typical error :  1. line: "hwme  NMI"
                 2. line: " 0x0800"
```

wrong expression type - integer expression expected

unexpected EOL - username for OTME event expected

unexpected EOL - An of the user given name for OTME event is expected.
Normally name is forgotten respectively name is written in the next line.

```
typical error:  1. line:  "OTME  "
                2. line:  "taskA  0x1234  "
```

separating BLANK expected

reserved name used

The use of names reserved from the system as names for OTME event is forbidden.

```
E.g. "otme  TRUE  0x1234"
```

name expected

The name of the OTME event is expected.

unexpected EOL - value for OTME event expected

unexpected EOL - The value of the defined OTME event is expected. Frequently the expression is forgotten respectively stands in the next line.

```
typical error :  1. line: "otme  taskA"
                 2. line: " 0x1234"
```

wrong expression type - integer expression expected

'/' expected

Unitnames for OTME events have to begin with prefix '/' like an option.
`E.g. "otme  taskA  0x1234  +DMA"`

unitname expected

The given name isn't an unitname for the OTME event. The names are CPU specific.
`E.g. "otme  taskA  0x1234  /DBM"`

name expected

The name of the timing event is expected.

wrong expression type - time expression expected

unexpected EOL - username for time event expected

unexpected EOL - An of the user given name for time event is expected.
Normally name is forgotten respectively name is written in the next line.

```
typical error:  "timecounter  "
                "unvalid_time  1ms--5ms  "
```

reserved name used

The use of names reserved from the system as names for timecounter event is forbidden.
`E.g. "timecounter  TRUE  1ns"`

separating BLANK expected

delaycounter name must be DELAY

The name for the delay counter event must be DELAY. No user chosen name could be used (only HAC32).

```
E.g. "timecounter  waiting  1000ns"     wrong
     "timecounter  DELAY    1000ns"      ok
```

name expected

wrong expression type - event expression expected

The given expression has not the type INTEGER or RANGE.

unexpected EOL - username for event expected

unexpected EOL - An of the user given name for the count event is expected.Normally name is forgotten respectively name is written in the next line.

```
typical error:  1. line:  "eventcounter  "
                2. line:  "unvalid_path  0--5  "
```

reserved name used

The use of names reserved from the system as names for counter event is forbidden.

```
E.g. "eventcounter  TRUE  0xff"
```

separating BLANK expected

unexpected EOL - username for externcounter expected

unexpected EOL - An of the user given name for the extern count event is expected.
Normally name is forgotten respectively name is written in the next line.

```
typical error:  1. line:  "externcounter  "
                2. line:  "unvalid_occur  5  "
```

separating BLANK expected

reserved name used

The use of names reserved from the system as names for externcounter event is forbidden.

```
E.g. "externcounter  TRUE  0xff"
```

name expected

wrong expression type - extern expression expected

The given expression has not the type INTEGER.

name expected

unexpected symbol - An of the user given name for the flag is expected.
Normally a writting error occurred.

```
e.g. "flags , motor_on",  "flags .motor_on"
```

name expected

unexpected symbol - A user given name for the flag is expected.
Normally a writting error occurred.

```
e.g. "flags motor_on, !LED_on"
                      ^         error
```

unexpected EOL - username for flag expected

unexpected EOL - An of the user given name for the flag is expected.
Normally name is forgotten respectively name is written in the next line.

```
typical error :  1. line:  "flags  "
                 2. line:  "unvalid_path   "
      or         1. line:  "flag unvalid_path , "
                 2. line:  "     motor_on"
```

reserved name used

The use of names reserved from the system as names for flags is forbidden.E.g. "flags TRUE "

separating BLANK expected

unexpected EOL - "." expected

unexpected EOL - Normally "." is forgotten respectively it stands in the next line.

```
typical error :  1. line:  "dlatch  "
                 2. line:  ".b extern_started 0xff  "
```

"." expected

unexpected EOL - name of channel B expected

unexpected EOL - Normally the channel name is forgotten respectively it stands in the next line.

```
typical error :  1. line:  "dlatch.  "
                 2. line:  "b extern_started 0xff  "
```

name of channel B expected

unexpected EOL - datalatch event name for dlatch event or blank expected

unexpected EOL - Datalatch event name for dlatch event or blank is expected.Normally the name is forgotten respectively it stands in the next line.

```
typical error :  1. line:  "dlatch.b  "
                 2. line:  "extern_started 0xff  "
```

separating BLANK expected

Frequently a wrong symbol is written instead of the dlatch event name.

```
typical error :  "dlatch.b ,extern_started 0xff"
                           ^
```

reserved name used

The use of names reserved from the system as names for dlatch event is forbidden.

```
E.g. "dlatch.b  TRUE  0xff"
```

name expected

unexpected EOL - data expression expected

unexpected EOL - The value of the defined dlatch event is expected. Frequently the expression is forgotten respectively stands in the next line.

```
typical error :  1. line:  "dlatch.b  extern_started"
                 2. line:  " 0xff"
```

separating BLANK expected

Frequently the value of the dlatch event is omitted or a wrong symbol is written instead of the dlatch event.

```
typical error :  "dlatch.b extern_started  ,0xff"
                                           ^         error
```

data expression expected

The value of the defined dlatch event is expected. Frequently the expression is from a wrong type or it's forgotten respectively a unexpected symbol stands in before the expression.

```
typical error :  "dlatch.b  extern_started  p:0xff"    or
                 "dlatch.b  extern_started  ,12"
```

data expression expected

The next value of the defined dlatch event is expected behind the written ",". Frequently the expression is from a wrong type, it's forgotten or the expression stands in the next line respectively a unexpected symbol stands in before the expression.

```
typical error :  "dlatch.b  extern_started  20 , p:0xff"    or
                 "dlatch.b  extern_started  20, "           or
typical error :  1. line:  "dlatch.b  extern_started  20,"
                 2. line:  " 0xff"
```

"." expected

The prefix "." for the subcommand from the aux command is expected. Normally the dot is forgotten respectively subcommand is written in the next line.

```
typical error :  1. line:  "aux a if cnt1", "aux .a if cnt1"
        or       1. line:  "aux "
                 2. line:  ".a if cnt1"
```

invalid subcommand

subcommand is omitted or given subcommand is not correct

```
e.g.:   "aux.  IF CNT1", "aux.as IF CNT1" or "aux. a IF CNT1"
```

unexpected EOL - "." expected

unexpected EOL - The prefix "." for the subcommand from the aux command is expected. Normally the dot is forgotten respectively subcommand is written in the next line.

```
typical error :  1. line:  "aux"
                 2. line:  ".a if cnt1"
```

unexpected EOL - subcommand expected

unexpected EOL - An subcommand for the aux command is expected.
Normally subcommand is forgotten respectively subcommand is written in the next line.

```
typical error :  1. line:  "aux."
                 2. line:  "a if cnt1"
```

unexpected EOL - "." expected

unexpected EOL - The prefix "." for the subcommand from the break command is expected. Normally the dot is forgotten respectively subcommand is written in the next line.

```
typical error :  1. line:  "break"
                 2. line:  ".trace if cnt1"
```

"." expected

The prefix "." for the subcommand from the break command is expected. Normally the dot is forgotten respectively subcommand is written in the next line.

```
typical error :  1. line:  "break trace if cnt1", "break .trace if cnt1"
        or       1. line:  "break "
                 2. line:  ".trace if cnt1"
```

unexpected EOL - subcommand expected

unexpected EOL - An subcommand for the break command is expected.
Normally subcommand is forgotten respectively subcommand is written in the next line.

```
typical error :  1. line:  "break."
                 2. line:  "trace if cnt1"
```

invalid subcommand

subcommand is omitted or given subcommand is not correct

```
e.g.:   "break.  IF CNT1", "break.trcae IF CNT1" or "break. trace IF CNT1"
```

"." expected

invalid subcommand

unexpected EOL - "." expected

unexpected EOL - subcommand expected

unexpected CONTINUE

Due to hardware restrictions it isn't possible to use in the global level or in the highest level of the trigger program the command CONTINUE.

invalid subcommand

subcommand is omitted or given subcommand is not correct

```
e.g.:   "c. CNT1 IF INT", "c.onn CNT1 IF INT" or "c.o n CNT1 IF INT"
```

unexpected EOL - username for counter event expected

unexpected EOL - An of the user given name for the counter event (TIMECOUNTER, EVENTCOUNTER, EXTERNCOUNTER) is expected.
Normally name is forgotten respectively name is written in the next line.

```
typical error :  1. line:  "counter.increment  "
                 2. line:  "interrupt_events   "
```

name expected

separating BLANK expected

unexpected EOL - counter subcommand expected

unexpected EOL - A subcommand of the counter command is expected.
Normally name is forgotten respectively name is written in the next line.

```
typical error :  1. line:  "count."
                 2. line:  "enable interrupt_events if INT1"
```

no more counter names expected

Due to hardware restrictions only one counter event name could be used in combination with this counter command (HAC32).

```
e.g.    counter.increment cnt_event1 cnt_event2   if  AB
                                     ^                    error
```

"." expected

invalid subcommand

name expected

separating BLANK expected

Frequently the flagname is forgotten respectively a wrong symbol is written.

```
typical error :  "flag.true , interrupt_occurred"
```

unexpected EOL - "." expected

unexpected EOL - subcommand for flag command expected

unexpected EOL - An subcommand for the flag command is expected.
Normally subcommand is forgotten respectively subcommand is written in the next line.

```
typical error :  1. line:  "flag.  "
                 2. line:  "true interrupt_occurred"
```

unexpected EOL - username for flag expected

unexpected EOL - An of the user given name for the flag event is expected.Normally name is forgotten respectively name is written in the next line.

```
typical error :  1. line:  "flag.true  "
                 2. line:  "interrupt_occurred"
```

unexpected EOL - subcommand for latch command expected

unexpected EOL - An subcommand for the latch command is expected.
Normally subcommand is forgotten respectively subcommand is written in the next line.

```
typical error :  1. line:  "latch."
                 2. line:  "enable if cnt1"
```

invalid subcommand

subcommand is omitted or given subcommand is not correct

```
e.g.:   "latch.  IF CNT1", "latch.eanble IF CNT1" or "latch. enable IF CNT1"
```

"." expected

invalid subcommand

unexpected EOL - "." expected

unexpected EOL - subcommand expected

"." expected

invalid subcommand

unexpected EOL - "." expected

unexpected EOL - subcommand expected

invalid subcommand

subcommand is omitted or given subcommand is not correct

```
e.g.:   "perf.  IF CNT1", "perf.enable IF CNT1" or "perf. enable IF CNT1"
```

unexpected EOL - subcommand for perf command expected

unexpected EOL - An subcommand for the perf command is expected.
Normally subcommand is forgotten respectively subcommand is written in the next line.

```
typical error :  1. line:  "perf."
                 2. line:  "enable if cnt1"
```

invalid subcommand

subcommand is omitted or given subcommand is not correct

```
e.g.:   "sample.  IF CNT1", "s.eanble IF CNT1" or "s. enable IF CNT1"
```

unexpected EOL - subcommand for sample command expected

unexpected EOL - An subcommand for the sample command is expected.
Normally subcommand is forgotten respectively subcommand is written in the next line.

```
typical error :  1. line:  "sample."
                 2. line:  "enable if cnt1"
```

"." expected

invalid subcommand

unexpected EOL - "." expected

unexpected EOL - subcommand expected

loopvariable "?" not allowed in expression

value with max. 45 bit size expected

Event/Time [range] values with max. 45 bitvalues are possible.

time value zero not permitted

Only time values with bigger than zero are possible.

DATA range borders must have 8 bit value

The values of the range borders can't exceed byte values (0..255).

```
e.g.  DATA.W0 errorvalues 0xe00--0xf11                    wrong
      DATA.B  ascii       '0'--'1'||'a'--'z'||'A'--'Z'    ok
```

value 0 not allowed

For HWME only values between 0x0001 and 0xffff are possible.

value with max. 16 bit size expected

only HWME || OTME values with max. 16 bitvalues are possible.

value with max. 32 bit size expected

only OTME values with max. 32 bitvalues are possible.

byte value expected in DATA.B declaration

value with max. word size expected (DATA.W)

TRIG or DLATCH value must be byte value

TRIG or DLATCH range borders must have byte values

only byte range at B0, B1, B2 and B3 allowed

only byte range at B, B0 and B1 allowed

value too big (max. 3 byte value)

value too big (max. 3 byte value)

byte value expected in DATA.B declaration

value with max. word size expected (DATA.W)

value too big (max. 3 byte value)

TRIG value must be bytevalue (only byte hexmask)

value with max. doubleword size expected

Time value larger than 32 bit.

value with max. word size expected

Only timerange values with max. 16 bitvalues are possible.

value with max. word size expected

Only eventrange values with max. 16 bitvalues are possible.

only simple ranges are allowed

Only event ranges with 2 borders are possible.

only simple ranges are allowed

Only extern ranges with 2 borders are possible.

value with max. word size expected

Only extern event range values with max. 16 bitvalues are possible.

DATA range borders must be limited by max. byte value (DATA.B)

DATA range borders must be limited by max. word value (DATA.W)

only byterange at B, B0, B1, B2 and B3 allowed

value with max. 39 bit size expected

Time value larger than 39 bit.

value with max. 39 bit size expected

Only timerange values with max. 39 bitvalues are possible.

value bigger than 65.535ms

Time value bigger than 65.535ms.

no timerange possible

Only simple time values allowed (HAC32 has only one time counter).

no eventrange possible

Only simple event values allowed. Ranges couldn't be realized (HAC32).

value with max. word size expected

Event value bigger than 16 bit or negative.

value with max. 48 bit size expected

Time value bigger than 48 bit.

value with max. 48 bit size expected

Only timerange values with max. 48 bitvalues are possible.

no bitmask or hexmask expected

Only address and integer values are allowed with the used breakpoint type (name).
Masks could be used only in combination with the OD breakpointtypes (like OD, ODH, ODHX, ODL, ODLX, ODX).

```
e.g.  ADDRESS OAR 0x12XX      // error
      ADDRESS OD  0x12XX      // ok
```

address value bigger than <bitnumber> bits

The address value exceeds the biggest address for the given operand breakpoint type.

```
e.g.  ADDRESS OD  0x12XXX     // error - at maximum 16bitvalue allowed for OD
      ADDRESS OD  0x12XX      // ok
```

only one addressvalue expected

Only one addressvalue or one addressrange could be assigned to the special bondout address events.

```
e.g. "address OD  0x1234 0x5678"

instead of

     "address OD  0x1234"
     "address ODX 0x5678"
```

value too big (max. 4 byte value)

value with max. 39 bit size expected

Event/externcounter value bigger than 39 bit.

value with max. 39 bit size expected

Event/externcounter range values with max. 39 bitvalues are possible.

value with max. 48 bit size expected

Event/externcounter value bigger than 48 bit.

value with max. 48 bit size expected

Only event-/externcounter range values with max. 48 bitvalues are possible.

value with max. 32 bit size expected

Only externcounter event value with max. 32 bitvalues are possible.

only bytewise hexmasks possible

Inside the hexmasks it's only bytewise "don't care" permitted.

```
e.g. DATA.L0 wrong_value    0Xx2345678
                               ^               error position
     DATA.L0 correct_value  0Xxx345678            ;  OK
     DATA.T0 correct_value  0X1fxxxx              ;  OK
     DATA.W1 correct_value  0Yxxxxxxxx10101011    ;  OK
     DATA.W0 wrong_value    0Yxxxxx1xx10101011
                                   ^              error position
```

no ranges allowed

Ranges couldn't be used as data event value. Please use hexmasks instead if possible (only bytewise "don't care").

```
e.g. DATA.B0 wrong_value    1--24
                            ^               error position
```

value with max. 28 bit size expected

Only event values with max. 28 bitvalues (28bit maxvalue - 8) are possible.

value with max. 28 bit size expected

Time value bigger than 28 bit (28bit maxvalue - 800.ns).

value 0 not permitted

Only time or event counter value > 0 expected.

not enough memory (for expression compiler)

Not enough system memory for the generation of the expression compiler formula.

only constant boolean expressions allowed

trigger level <*labelname*> doesn't exist

Not existing level labelname used in GOTO command.

data respectively vdata event <*name*> declared but not used

trigger or dlatch event <*name*> declared but not used

TIMECOUNTER event <*name*> declared but never used at all

TIMECOUNTER event <*name*> declared but not set

It's only a warning which could give a hint about a possible programming error.

In a trigger program each counter event is normally used as input as well as output. The command counter.increment or counter.on wasn't used for timecounter event delay.

```
e.g.        EVENTCOUNTER skip_first_100_cycles    100.
            TIMECOUNTER  delay                   100ms
     ll0:   counter.increment skip_first_100_cycles
            continue                           if  skip_first_100_cycles
     ll1:   sample.enable
            break.trace                        if  delay
```

TIMECOUNTER event <*name*> declared but not used in condition

This message is an error only in combination with the HAC32. With other analyzer hardware it's only a warning which could give a hint about a possible programming error.

In a HAC32 trigger program each used counter event must be used at least once in a trigger condition and must be enabled with the command Counter.Increment in the same level !

```
e.g.        EVENTCOUNTER skip_first_100_cycles    100.
            TIMECOUNTER  delay                   100ms
     ll0:   counter.increment skip_first_100_cycles
            continue                           if  skip_first_100_cycles
     ll1:   sample.enable
            counter.increment delay
            break
```

TIMECOUNTER event <*name*> set but not used in condition

It's only a warning which could give a hint about a possible programming error.

In a trigger program each counter event is normally used as input as well as output. The timecounter event delay wasn't used inside a condition.

```
e.g.        EVENTCOUNTER skip_first_100_cycles    100.
            TIMECOUNTER  delay                   100ms
      ll0:  counter.increment skip_first_100_cycles
            continue                          if  skip_first_100_cycles
      ll1:  sample.enable
            counter.increment delay
            break.trace
```

TIMECOUNTER event <*name*> used but not set

This message is an error only in combination with the HAC32. With other analyzer hardware it's only a warning which could give a hint about a possible programming error.

In a HAC32 trigger program each defined counter event must be enabled with command Counter.Increment countername in the level in which it is used as an input event in a trigger condition.

```
e.g.        EVENTCOUNTER skip_first_100_cycles    100.
            TIMECOUNTER  delay                   100ms
      ll0:  counter.increment skip_first_100_cycles
            continue                          if  skip_first_100_cycles
      ll1:  sample.enable
            break                             if  delay
```

EVENTCOUNTER counter <*name*> declared but never used at all

EVENTCOUNTER counter <*name*> declared, but not used in condition

This message is an error only in combination with the HAC32. With other analyzer hardware it's only a warning which could give a hint about a possible programming error.

In a HAC32 trigger program each used counter event must be used at least once in a trigger condition and must be enabled with the command Counter.Increment in same level !

```
e.g.        EVENTCOUNTER skip_first_100_cycles    100.
            EVENTCOUNTER delay                   1000.
      ll0:  counter.increment skip_first_100_cycles
            continue                          if  skip_first_100_cycles
      ll1:  sample.enable
            counter.increment delay
            break
```

EVENTCOUNTER counter <*name*> used in condition, but not set

It's only a warning which could give a hint about a possible programming error.

In a trigger program each counter event is normally used as input as well as output. The command counter.increment wasn't used for event event delay.

```
e.g.        EVENTCOUNTER skip_first_100_cycles     100.
            EVENTCOUNTER delay                    1000.
      ll0:  counter.increment skip_first_100_cycles
            continue                          if  skip_first_100_cycles
      ll1:  sample.enable
            break.trace                       if delay
```

FLAG <*name*> declared but not used

EVENT counter <*name*> declared but not used in condition

It's only a warning which could give a hint about a possible programming error.

In a trigger program each counter event is normally used as input as well as output. The command counter.increment wasn't used for event event delay.

```
e.g.        EVENTCOUNTER skip_first_100_cycles     100.
            EVENTCOUNTER delay                    1000.
      ll0:  counter.increment skip_first_100_cycles
            continue                             if  skip_first_100_cycles
      ll1:  sample.enable
            counter.increment delay
            break.trace
```

label START does not exist

Jump to the n o t existing level START.

:Sprung zu nicht existierender Ebene mit dem label START entdeckt

FLAG <*name*> used but not never set

address event <*name*> declared but not used

FLAG <*name*> set but never used as input event

data events used in too many level

Data events can be used only in the first 4 levels. If data events are used in global commands then at most 4 levels can be used.

The HAC32 has only 1 data event for each of the first 2 (construction 1) respectively 1 (construction 2) levels and data events aren't allowed in global commands.

trigger or dlatch events used in too many levels

Trigger or dlatch events can be used only in the first 16 levels. If trigger respectively dlatch events are used in global commands then at most 16 levels can be used.

The HAC32 has only 1 trigger event for each of the first 2 (construction 1) respectively 1 (construction 2) levels and trigger events aren't allowed in global commands.

TIMECOUNTER event <*name*> set, but not used in condition

It's only a warning which could give a hint about a possible programming error.

```
e.g. normal programming example
            EVENTCOUNTER skip_first_100_cycles    100.
            TIMECOUNTER  delay                    100ms
      ll0:  counter.increment skip_first_100_cycles
            continue                             if  skip_first_100_cycles
      ll1:  sample.enable
            counter.increment delay
            break                                if  delay
```

too many flags used

At most a number of 2 flags (ANAICD) can be used.

EVENTCOUNTER condition <*name*> used, but never set

This message is an error only in combination with the HAC32. With other analyzer hardware it's only a warning which could give a hint about a possible programming error.

In a HAC32 trigger program each defined counter event must be enabled with the command Counter.Increment countername in the level in which it is used as an input event in a trigger condition.

```
e.g.       EVENTCOUNTER skip_first_100_cycles    100.
           EVENTCOUNTER delay                    1000.
     110:  counter.increment skip_first_100_cycles
           continue                              if  skip_first_100_cycles
     111:  sample.enable
           counter.increment delay
           break                                 if  delay
```

too many input events - no free MUX for <*eventname*>

Too many global input events exist.

The global instructions are "made" in every level (it seems they were done as local one; they have the same meaning as the local) and therefore they have the same restrictions as the local instructions. 4 input events are possible in each level.

too many counter used

At most a number of 5 counters can be used. 3 counters of them are universal counters for the usage as timecounter event or counter events which are used in conditions. The other 2 counters can be used only as counter events which can't be used in conditions.

too many local input events - in level <*levelname*> no free MUX for <*eventname*>

Too many local input events exist. The number of input events in each level is the sum of the local used input events plus the global used input events.4 input events are possible in each level.

too many flags used

At most a number of 8 (3 in HA120 version) flags can be used. Therefrom the number of the used counter and a possibly existing TRACE recording switch (sample.enable - not in HA120 version) has to be subtracted.

too many universal counter used

At most a number of 3 (2 in ECC8 version) universal counter can be used. They must be always used for timecounter events and for counter events which are used in conditions.

HWME event <*name*> declared but not used in condition

It's only a warning which could give a hint about a possible programming error.

In a trigger program each HWME event is normally used as input. The actual hardware event message wasn't used inside a condition.

```
e.g.       HWME NMI 0x0800
     ll0:  counter.increment skip_first_100_nmi  if  NMI
           continue                             if  skip_first_100_nmi&&NMI
     ll1:  sample.enable
           counter.increment delay
           break.trace
```

EXTERNCOUNTER event <*name*> declared but not used

OTME event <*name*> declared but not used in condition

It's only a warning which could give a hint about a possible programming error.

In a trigger program each OTR event is normally used as input. The actual hardware event message wasn't used inside a condition.

```
e.g.       OTME searched_otme_value 0x1234
     ll0:  sample.enable  if  searched_otme_value
```

EXTERNCOUNTER event <*name*> used but not set

FLAG <*name*> set, but not used

EXTERNCOUNTER event <*name*> set, but not used in condition

no counter for EXTERNCOUNTER event <*name*> free

EVENTCOUNTER event <*name*> set, but not used in condition

This message is an error only in combination with the HAC32. With other analyzer hardware it's only a warning which could give a hint about a possible programming error.

In a HAC32 trigger program each used counter event must be used at least once in a trigger condition in the level in which the counter was enabled !

```
e.g.       EVENTCOUNTER skip_first_100_cycles    100.
           EVENTCOUNTER delay                   1000.
     ll0:  counter.increment skip_first_100_cycles
           continue                           if  skip_first_100_cycles
     ll1:  sample.enable
           counter.increment delay
           break                              if  delay
```

level <*name*> defined, but never used or unreachable

timecounter events could be used only in level 1 respectively 2

Due to hardware restrictions time counters could be used only in the last level (HAC32).

For construction 1 the last level is level 2.

For construction 2 the last level is level 1.

too many delay counter used in level <*number*>

At most a number of 1 delay counter can be used (HAC32).

too many event counter used in level <*number*>

General at most a number of 1 event counter can be used in each level in construction 1 (HAC32). In construction 2 it is possible to use in level 0 2 event counter and in level 1 one event delay counter.

event counter <*name*> never used in condition

```
e.g.   EVENTCOUNTER skip_cycles  100.
       LL0:  Counter.Increment skip_cycles  if  AB&&WRITE
             Sample.Enable                  if  AB&&WRITE
       ^                                                 error
correct:
e.g.   EVENT skip_cycles  100.
       LL0:  Counter.Increment skip_cycles  if  AB&&WRITE
             Sample.Enable                  if  AB&&WRITE&&skip_cycles
```

level <*name*> not reachable - CONT in level before is missed

In level before it isn't used any CONTinue command, but this is obligatory.

```
e.g.  START:  sample.enable      if ab&&write
      LEVEL2: counter.increment DELAY
              break              if DELAY
```

The line

```
              continue           if ab&&write
```

must be added in level START to get a correct trigger program.

too many time counter used

At most a number of 2 time counter can be used in the analyzer programming (FIRE).

too many event counter used

At most a number of 2 event counter can be used in the analyzer programming (FIRE).

too many data events used

At most a number of 2 data event depending on the hardware release can be used in the analyzer programming (FIRE). Each used VDATA events will reduce the number of available DATA events.

counter use in 2 different levels

Due to hardware restrictions a counter could be used only inside one level in commands (like COUNTER.INCREMENT) and in conditions as input event (HAC32).

```
e.g.  START:  sample.enable           if ab&&write
              continue                if ab&&write&&cnt0
      LEVEL2: counter.increment cnt0  if bb&&read
```

not declared name used

All names which could be freely created from the user have to be declared. The only exception are the labels. All data, trigger, time and counter events and the flags must be declared. Predefined system names are used for address events and that's the reason why no declaration is necessary.

too many data events used

At most 2 data events are possible in each level. Every global used data event reduces the number of the local (in a level) usable.

too many counter events used

At most 3 counter events are possible in each level. Every global used counter event reduces the number of the local (in a level) usable.

too many HWME events used

At most 2 HWME events are possible.

too many trigger events used

At most 3 trigger events are possible in each level and trigger channel (A respectively B). Every global used trigger event reduces the number of local (in a level) usable.

only 2 data events possible

too many OTME events used

At most 2 OTME events are possible.

too many trigger events used

ECC8: At most 2 trigger events are possible in each level. Every global used trigger event reduces the number of local (in a level) usable.
HA120: At most 1 trigger event is possible in each level. Every global used trigger event reduces the number of local (in a level) usable.
HAC: construction 1 (c1): 1 trigger event could be used in each of the first 2 levels.
construction 2 (c2): 2 trigger events could be used in the first level
If a trigger event is used negated as well as not negated in two different conditions then it counts as two trigger events !!!
This need of resources differs from other analyzer trigger units like HA120, ECC8. They need only one trigger event for the same trigger program !

```
e.g.        trig.a one 1
     LL0:  sample.enable if   one     ; t w o   trigger events used !
           break         if  !one
```

too many dlatch events used

At most 1 (HA120) dlatch events is possible in each level. Every global used dlatch event reduces the number of local (in a level) usable.

too many data events used

At most 2 global data events are in the whole trigger program possible. The effective number depends on the hardware release.

no trigger event allowed in this level

Due to hardware restrictions of the HAC32 it isn't possible to use trigger events in the third level (c1) respectively second level in construction 2 (c2). Only in the first 2 respectively 1 levels are trigger events allowed (c1 respectively c2).

level name is declared twice

too many levels defined

At most 64 (SA120), 8 (HA120), 4 (ECC8), 4 (FIRE) respectively 8 (FIRE) or 4 (ICD) levels are possible. The actually used number of level exceeds these limits.

unexpected level name START

The level name START could be used only for the first level (only HAC32).

```
e.g. timecounter DELAY 100ms
     WAITING:  sample.enable if read&&ab
               cont          if read&&ab
     START:    trigger.spot  if DELAY
               goto WAITING  if DELAY
     ^                                   error
```

Correction: just exchange WAITING and START or use a different name for START.

level name unexpected

In a HAC32 trigger program could be used either only global commands or local commands but **not global commands together with local commands in a level** !

```
e.g.      eventcounter   writes_cnt
          address        ab         flags
          counter.increment writes_cnt    if   write&&ab
          level0:   sample.enable         if   write&&ab
          ^                                            error
          level1:   trig.a                if   nmi
```

counter <*name*> not declared

counter <*name*> not declared

countername expected

The given name isn't a counter event.

All names can only exist with one meaning and must be different from all keywords.

```
e.g. FLAG        engine_on
     TIMECOUNTER delay     100.ms
     counter.increment engine_one, delay  if  write&&ab
                       ^                                 error
```

counter event name DELAY unexpected

Due to hardware restrictions it is not possible to use the counter event DELAY with the command Counter.Restart (HAC32).

flag not declared

flagname expected

The given name isn't a flag name.

All names can only be exist with one meaning and must be different from all keywords.

```
e.g. FLAG        engine_on
     TIMECOUNTER delay 100ms
     Flag.TRUE engine_on, delay  if write&&ab
                          ^                    error
```

not enough memory (for internal table)

name has multiple meaning

address event <*addr_name*> is declared multiple

The address event with the name addr_name has multiple declarations. This is not permitted. Please use the second address eventtype if possible.

```
e.g.      "address OD  0x10"
          "address OD  0x30"
                   ^                error position
correct:
          "address OD  0x10"
          "address ODX 0x30"
```

address event isn't possible

The given address event isn't available with the current analyzer hardware.

address event <*addr_name*> is multiple declared

The address event with the name addr_name has multiple declarations. This causes a special treatment of the declarations. All these declarations will be treated as one declaration (combined automatically).

```
e.g.      "address AB   sp:100  ud:200        "        are
          "address AB   sd:300  v.range(flags)"
equal to "address AB   sp:100  ud:200  sd:300  v.range(flags)"
```

address event <*addr_name*> is already declared as <*addr_name2*>

The address event is used in a declaration under a different name already before. This is not permitted. Please use the second address eventtype if possible.

```
e.g.      "address OD   0X1234"
          "address ODL  0X56  "
                    ^              error position
correct: "address OD   0x1234"
          "address ODLX 0x56  "
```

This 2 different address events use the same physical breakpoint type.

no levelname

The given name is no levelname.

unexpected level name START

The level name START could be used only for the first level (only ICD).

```
e.g. selector    ab    P:1000 /data.byte 0x77 /read
     timecounter DELAY 100ms
     WAITING:  sample.enable if ab
               continue      if ab
     START:    trigger.pulse if DELAY
               goto WAITING  if DELAY
     ^                                  error
```

Correction: just exchange WAITING and START or use a different name for START.

not enough memory (for internal table)

flag <*name*> not declared

flagname expected - <*name*> has wrong event type

The given name isn't a flag name.
All names can only be exist with one meaning and must be different from all keywords.

```
e.g. FLAG        engine_on
     TIMECOUNTER delay 100ms
     Flag.TRUE engine_on, delay  if write&&ab
                          ^                    error
```

not enough memory (for internal table)

either IN0 or IN1 can be used in the analyzer trigger program

only one TCODE event can be used in the analyzer trigger program

A different TCODE input event was already used in the trigger programming. In the whole trigger program only one TCODE input event could be used.

too many input events

Too many global input events exist.

The global instructions are "made" in every level (it seems they were done as local one; they have the same meaning as the local) and therefore they have the same restrictions as the local instructions. 4 input events are possible in each level.

too many local input events

Too many local input events exist. The number of input events in each level is the sum of the local used input events plus the global used input events.4 input events are possible in each level.

too many different conditions

Due to hardware restrictions (HAC32) it isn't possible to realize the additional trigger condition. Too many different conditions are used in the actual level before.

```
e.g.    sample.enable                       if AB
        counter.increment  variable_reads   if write&&BB
        mark.b                              if read&&AB
                                               ^  error
```

This error could appear even if only 2 conditions are used in a level.

```
e.g.    counter.increment  variable_reads   if write&&BB
        mark.b                              if write&&BB&&counter_event2
                                                        ^   error
```

The same triggerpoint TP1 must be used for both different conditions. The first condition will be combine implicit the counter event variable_reads with write&&BB.

condition not permitted with these commands

In the actual line commands are used which couldn't be combined with such a condition. Only the command Sample.Enable is allowed with a condition construction "if !(..)". This restriction only exists for HAC32.

```
e.g.    sample.enable, mark.a  if  !(AB&&read)
                       ^                          error
```

the counter increment couldn't be depending from a counter input event

The command Counter.Increment couldn't be used in combination with a counter event in the condition (only HAC32).

```
e.g.    eventcounter cnt_100times  100.
        address      ab            v.range(flags)
        counter.increment cnt_100times, mark.a  if  AB&&read&&cnt_100times
                                                              ^   error
```

not enough resources for AB or BB input event

Due to hardware restrictions (HAC32) it isn't possible to realize the additional trigger condition. Too many different conditions are used in the actual level before.

```
e.g.    sample.enable                          if BB
        counter.increment  variable_reads      if write&&BB
        out.a                                  if AB
        mark.a                                 if read&&AB
```

unexpected trigger event in condition

Due to hardware restrictions (HAC32) it isn't possible to combine in this case (too many different conditions in this level) the external trigger events (e.g. TRIG.A Bankno0 00) with the command MARK.B or the input event BB.

command or condition not realizable in current level

Due to hardware restrictions (HAC32) it isn't possible to use in this level the command mark.a or in the condition the input event AB respectively a trigger event (e.g. TRIG.A Bankno0 00).

This could be realized only in level 0.

command and condition combination couldn't be realized

Due to hardware restrictions (HAC32) it isn't possible to realize with one trigger point the given command and condition combination.
The commands mark.a and mark.b couldn't be combined in one line.

```
e.g.  MARK.A, MARK.B  IF  AB&&WRITE
```

The input event AB and/or a trigger event (e.g. TRIG.A Bankno0 00) couldn't be combined with the input event BB.

```
e.g.  SAMPLE.ENABLE   IF  AB&&BB
```

The command MARK.A couldn't be combined with the input event BB.
The command MARK.B couldn't be combined with the input event AB or a trigger event.
Sometimes it is possible to split the command and condition combination into two combinations.

counter name expected in condition

Due to hardware restrictions (HAC32) the command Counter.Restart countername must be used in combination with the condition countername and without any other input event (AB, WRITE, ...).

```
e.g.    COUNTER.RESTART skip_cycle_number  if  skip_cycle_number
                                               ^                    OK

        COUNTER.RESTART skip_cycle_number  if  second_counter
                                               ^                    error
```

only counter name in condition expected

At least one input event is used in the condition which is different to a counter event name (like AB, WRITE, ...).

Due to hardware restrictions (HAC32) the command Count.Restart countername must be used in combination with the condition countername.

```
e.g.    COUNTER.RESTART skip_cycle_number  if  skip_cycle_number&&READ
                                                                ^ error
```

counter name can't be used with this condition

Due to hardware restrictions (HAC32) it isn't possible to combine a counter name, which is already used in a first condition, with the command Counter.Increment and a different second condition.

```
e.g.   mark.a                    if ab&&read&&count0    // triggerpoint 0
       counter.increment count0  if bb&&write           // triggerpoint 1
                         ^                     error
```

counter name can't be used in 2 different conditions

Due to hardware restrictions (HAC32) it isn't possible to use a counter name, which is already used in a first condition, in a different second condition.

```
e.g.   mark.a if ab&&read&&count0                       // triggerpoint 0
       mark.b if bb&&read&&count0                       // triggerpoint 1
                          ^         error
```

MARK.A and MARK.B couldn't be used in the same line

Due to hardware restrictions (HAC32) it isn't possible to realize the two mark commands in this way. Please use instead two command lines.

```
e.g.   mark.a, mark.b, sample.enable if read
               ^                                  error
ok:    mark.a, sample.enable if read
       mark.b, sample.enable if read
```

Counter.Increment couldn't be used twice in the same line

Due to hardware restrictions (HAC32) it isn't possible to realize the two counter.increment commands in one line. In each level could be used only one counter and it must be in each level a different one.

```
e.g.   counter.increment cnt0, mark.a, counter.increment cnt1  if  read
                                       ^                error
```

command couldn't be used together with MARK.C

The command MARK.C could be used only without any other command in one single line (HAC32).

```
e.g.  mark.c, out.a  if  CB
              ^             error
```

The input event AB or BB is already used in a different mode negation before.

The input events AB respectively BB could be used in the whole analyzer trigger program only in a unique way. In any condition the input event has to be used always with or always without a logical not.

```
e.g.    sample.enable                if  AB&&read
        counter.increment  adr_reads if !AB&&read
                                        ^           error
```

condition N:CB not allowed

The current analyzer hardware doesn't support the negation of the input event CB.

```
e.g. mark.c if !CB
               ^     error
```

too many counter events in level

Due to hardware restrictions it is impossible to use a delay counter event named DELAY together with another counter event in the same level.
Delay counters could be used only in the last level.

```
e.g.       EVENTCOUNTER SKIP_READS 10.
           EVENTCOUNTER DELAY      123.
     LL0: sample.enable  if  SKIP_READS&&DELAY
                                    ^      error
or   LL0: sample.enable  if  SKIP_READS
          break          if  DELAY
                             ^                error
```

AB and BB not possible in one condition

Due to hardware restrictions it is impossible to use in a condition the two address events AB and BB at the same time.

```
e.g.  sample.enable if AB||BB
                          ^   error
```

Please use instead if possible two command lines like

```
      sample.enable if AB
      sample.enable if BB
```

only one cpu specific event in condition permitted

Due to hardware restrictions it is impossible to use in a condition two different cpu specific events at the same time.

e.g.

```
Level0: sample.enable  if  READ&&WRITE
                                ^          error
```

CB not used alone with MARK.C

The input event CB couldn't be used together with at least one other command than MARK.C (HAC32). The command MARK.C could be used only without any other command in one single line.

```
e.g.  out.a, mark.c  if  CB
      ^                    error
```

Currently it is no support implemented for using the input event CB with commands different from MARK.C !

e.g. OUT.A IF CB

condition !dsel_name not allowed

The analyzer hardware of HAC32 doesn't support the negation of a data selector event.

```
e.g.  data.b0 digits      '0'--'9'
      break   if  !digits
                  ^             error
```

The negation of the data selector values could be used instead in some cases as a workaround.

```
e.g.  data.b0 no_digit  !'0'--'9'
      break   if  no_digit
```

MARK.C not used with CB only

The command MARK.C could be used only with the input event CB (HAC32). No other input event could be combined with MARK.C in one single line.

```
e.g.  mark.c  if  AB
                  ^     error
```

delay counter expected in level <*name*> <>

Level 2 (construction 1) respectively level 1 (construction 2) could be used only in combination with a delay counter (HAC32).

mark.c command must be used in level <*n*> too

Due to hardware restrictions the mark.c command must be used either in no level or in every level (HAC32).

CONTinue command must be used in level 0 with each condition

Due to hardware restrictions the CONTinue command must be used in combination with each of the 2 trigger conditions of the level 0 (HAC32). This is only valid for the construction 2.

```
e.g.                          EVENTCOUNTER skip_writes 100.
                              ADDRESS      ab          v.range(flags)
                              ADDRESS      bb          motor_on
     waiting:                 continue                 if  read&&ab
                             ^                                      error
                              sample.enable            if  read&&ab
                              sample.enable            if  write&&bb
     skipping_first_writes:   goto waiting             if  skip_writes
                              counter.increment skip_writes
```

The line

```
                              continue                  if  write&&bb
```

must be inserted in level waiting too.

Trig.A command must be used in level 0 with each condition

Due to hardware restrictions the Trig.A command must be used in combination with each of the 2 trigger conditions of the level 0 (HAC32). This is only valid for the construction 2.

```
e.g.          ADDRESS  ab    v.range(flags)
              ADDRESS  bb    motor_on
     waiting: trig.a           if  read&&ab
              ^                                          error
              sample.enable    if  read&&ab
              sample.enable    if  write&&bb
```

The line

```
              trig.a           if  write&&bb
```

must be inserted in level waiting too.

BREAK or CONTINUE command must be used in level 0 with each condition

Due to hardware restrictions the BREAK command must be used in combination with each of the 2 trigger conditions of the level 0 (HAC32). This is only valid for the construction 2.

An used CONTINUE command will be converted automatically into a BREAK command if no DELAY counter is used.

```
e.g.          ADDRESS  ab    v.range(flags)
              ADDRESS  bb    motor_on
     waiting: break            if  read&&ab
              ^                                          error
              sample.enable    if  read&&ab
              sample.enable    if  write&&bb
```

The line

```
              break            if  write&&bb
```

must be inserted in level waiting too.

GOTO command must be used in level 0 with each condition

Due to hardware restrictions the GOTO command must be used in combination with each of the 2 trigger conditions of the level 0 (HAC32). This is only valid for the construction 2.

```
e.g.          ADDRESS  ab    v.range(flags)
              ADDRESS  bb    motor_on
     waiting: goto waiting     if  read&&ab
              ^                                          error
              sample.enable    if  read&&ab
              sample.enable    if  write&&bb
```

The line

```
goto waiting   if  write&&bb
```

must be inserted in level waiting too.

only <*n*> levels allowed <>

Too many levels used in trigger programming (HAC32).
Due to hardware restrictions the number of programming level depends on the programming construction.
In construction 1 are at maximum 3 levels possible.
In construction 2 are at maximum 2 levels possible.

only GOTO level 0 possible

Due to hardware restrictions it is impossible to goto a different level than level 0 (HAC32).
If the GOTO is only used to reach the next level the command GOTO should be replaced by the command CONTINUE.

```
e.g.  WAITING:  continue            if  write&&ab
      SAMPLING: sample.enable
                goto NEXT           if  write&&bb
                     ^                                  error
      NEXT:     sample.enable
                counter.increment DELAY
                goto WAITING        if  DELAY

e.g. START:        sample.enable
                   continue           if  read&&AB
     SECOND_LEVEL: sample.enable
                   continue           if  write&&AB
     THIRD_LEVEL:  goto SECOND_LEVEL  if  DELAY
                        ^                           error
```

In this case it could be only level START.

last level command overwritten

The current command overwrites the previous level command (CONTinue, GOTO) in the same line.

counter event name DELAY unexpected

Due to hardware restrictions it is only possible to use the counter event DELAY in the last level (HAC32).
For construction 1 the last level is level 2.
For construction 2 the last level is level 1.

too many input events

Too many global input events exist.
The global instructions are "made" in every level (it seems they were done as local one; they have the same meaning as the local) and therefore they have the same restrictions as the local instructions.
At most 6 input events are possible in each level. The minimum is 3. The effective usable number depends on the size of the trace-memory and the number of the used levels.

too many local input events

Too many local input events exist. The number of input events in each level is the sum of the local used input events plus the global used input events.At most 6 input events are possible in each level. The minimum is 3. The effective usable number depends on the size of the trace-memory and the number of the used levels.

too many input events

Too many global input events exist.

The global instructions are "made" in every level (it seems they were done as local one; they have the same meaning as the local) and therefore they have the same restrictions as the local instructions. 6 input events are possible in each level.

too many local input events

Too many local input events exist. The number of input events in each level is the sum of the local used input events plus the global used input events.6 input events are possible in each level.

too many input events over premultiplexer

Too many input events exist which uses the premultiplexer. The number of input events in each level is the sum of the local used input events plus the global used input events.
Only 2 input events over the premultiplexer are possible in the whole trigger program.

TRIG.B events are locked at the moment

All TRIG.B events are locked at the moment. The use of Latch.Enable disables automatically the TRIG.B events. Only TRIG.A respectively DLATCH events are now available.

TRIG.B events are locked at the moment

The use of DLATCH events disables automatically the TRIG.B events. Either TRIG.B events or DLATCH events are usable, but not both at the same time.

LATCH.ENABLE not used

The dlatch event is used as an input event, but at no time the data will be latched with the command Latch.Enable. The result is: the dlatch event is always FALSE or TRUE (depending on his programming value).

too many special XA input events

Too many special input events are used.
The given input event couldn't be used in combination with another input events given before. The maximum number of 3 special events is exceeded (only ICE-XA).

data or trigger event in this level not permitted

Due to hardware restrictions (HAC) data and trigger events could be used only in the first 2 levels in program construction 1 (c1) or in the first level in program construction 2 (c2).

only one data, trigger or counter event in condition permitted

Due to hardware restrictions it is impossible to use in a condition two different data events at the same time. The same restriction is valid also for trigger and counter events.

```
e.g.    eventcounter first_cnt_event  100
        timecounter  second_cnt_event 200ms
Level0: sample.enable  if  READ&&AB&&first_cnt_event&&second_cnt_event
                                                    ^               error
```

data or trigger event is already used in a different negation mode before

A data or trigger event could be used in a level only in a unique way. In any condition inside a level the data or trigger event has to be used always with or always without a logical not.

```
e.g.   data  letter  'a'--'z' 'A'--'Z'
       sample.enable               if  letter&&read
       counter.increment adr_reads if !letter&&read
                                     ^            error
```

only one data, trigger or counter event in each level permitted

Due to hardware restrictions it is impossible to use in a level two different data events at the same time (HAC32). The same restriction is valid also for trigger and counter events (only construction 1).
The only exception from this rule is in construction 2 and level 0 are 2 counter events possible.

```
e.g.    data.b0 first_data_event  1
        data.b0 second_data_event 2
Level0: sample.enable  if  READ&&AB&&first_data_event
        sample.enable  if  READ&&AB&&second_data_event
                                    ^                 error
Level1: ...
Level2: ...
```

counter events couldn't be negated

A counter event couldn't be negated logical inside a condition as possible with e.g. AB (AlphaBreak).

```
e.g.   timecounter  delay  10.ms
       sample.enable  if  !AB&&!delay
                              ^          error
```

Please instead if possible a construction like

```
       sample.enable  if  !(AB&&delay)
```

no more MUX free in this condition

Due to hardware restrictions it is impossible to use in a condition two different events which need a MUX at the same time (HAC32).

no more MUX free in this level

The given input event needs a MUX, but all in this level available MUX are already used from different input events. Additional input events which need a MUX couldn't be realized.

input event not possible in actual level

The given input event couldn't be used in the actual level in a condition.

delay counter event only permitted in level <*n*> <>

Due to hardware restrictions it is only possible to use a delay counter event named DELAY in the last level (HAC32). The counting of the levels begins at 0.
construction 1 : only in level 2 allowed
construction 2 : only in level 1 allowed

```
e.g.              TIMECOUNTER  DELAY  1ms
     Level0:      sample.enable       if  READ&&AB
                  continue            if  READ&&AB
     SECOND:      sample.enable
                  continue            if  WRITE&&AB&&DELAY
                                                ^       error
                  counter.increment DELAY
     DELAYLEVEL:  ...
```

delay counter event DELAY expected <>

Only a delay counter event named DELAY could be used in level 1 respectively 2 in combination with the programming construction 2 respectively 1 (HAC32).Please rename your counter event with DELAY if possible. In the example below it is possible to replace just the counter name to avoid the error message.

```
e.g.                          EVENTCOUNTER  skip_writes   100.
                              ADDRESS  ab                 v.range(flags)
                              ADDRESS  bb                 motor_on
     waiting:                 continue                 if  read&&ab
                              continue                 if  write&&bb
                              sample.enable            if  read&&ab
                              sample.enable            if  write&&bb
     skipping_first_writes:   goto waiting             if  skip_writes
                                                           ^          error
                              counter.increment skip_writes
```

counter event <*name*> used in too many levels

Due to hardware restrictions one counter event could be used only in one level (HAC32). It couldn't be used in several levels in the same trigger program. In the same level the counter event could appear in conditions and commands like counter.increment repeated.

no suitable DSEL MUX free in this condition

Due to hardware restrictions it is impossible to use in the conditions two different events which needs the same special MUX at the same time (FIRE).Please try to use the DATA event as early as possible in the trigger program.

too many input events or no suitable MUX free in this condition

Only at maximum 10 special input events are possible in the whole program. Due to hardware restrictions not all input events could be selected with all MUX.

too many input events over premultiplexer or no suitable MUX free

Only at maximum 6 input events over the premultiplexer are possible in the whole trigger program. Due to hardware restrictions not all input events could be selected with all PREMUX.

breakpoint type <breakpointname> not set and for this always FALSE

The given breakpoint inside a condition isn't defined as an address event in the trigger program nor set outside the trigger program with the Break.Set command.

```
e.g.    address ab p:1000
        sample.enable  if  bb
                           ^        warning position
```

too many input events

too many global input events used - no more MUX free in the global level
The given input event needs a MUX, but all in this level available MUX are already used from different input events. Additional input events which need a MUX couldn't be realized.
The global instructions are "made" in every level (it seems they were done as local one; they have the same meaning as the local) and therefore they have the same restrictions as the local instructions.
At most 4 input events are possible in each level.

too many input events

too many local input events used - no more MUX free in this level
The given input event needs a MUX, but all in this level available MUX are already used from different input events. Additional input events which need a MUX couldn't be realized.
Too many local input events exist. The number of input events in each level is the sum of the local used input events plus the global used input events.4 input events are possible in each level.

breakpoint type not set and for this always FALSE

The given breakpoint inside a condition isn't defined as an address event in the trigger program nor set outside the trigger program with the Break.Set command.

```
e.g.    address ab p:1000
        sample.enable  if  bb
                           ^        warning position
```

not enough system memory

Not enough system memory for internal table.

not enough system memory

Not enough system memory for internal command table.

only one mode possible - other mode already selected

The input event ABX respectively BBX was already used in the trigger programming in a different mode. In the whole trigger program only one mode could be selected for ABX respectively BBX.

this input event couldn't be realized

The given input event couldn't be used in combination with another input event given before.

Please use instead 2 separate commands if possible.

```
e.g.    sample.enable  if  write&&int&&ab
                                 ^          error
```

use instead

```
        sample.enable  if  write&&ab
        sample.enable  if  int&&ab
```

mark <*mark*> command couldn't used in level <*n*> <>

Due to hardware restrictions not all mark commands could be used in all levels.
HAC32: MARK.A could be used only in level 0 (construction 1 and 2)
MARK.B could be used only in level 0 (construction 2)
MARK.B could be used only in level 1 (construction 1)
MARK.C could be used in level 0, 1 and 2

break or t.a command only permitted in highest level

Due to hardware restrictions it is only possible to use the break respectively the t.a command in the highest level of the trigger program. This is programming model independend.

only Sample.Enable command possible with the condition

Only the command Sample.Enable could be combined with the negated trigger condition. It isn't permitted to all other commands like Counter.Increment, Mark.A, Trigger.A.

```
e.g.  sample.enable, mark.a  if  !(ab&&write)
                     ^                          error
```

this input event couldn't be realized

The given input event couldn't be used in combination with another input event given before.
The input event uses the same premultiplexer which is already used by another input event (RUNF, CB, BUSA).

this input event couldn't be realized

The given input event couldn't be used in combination with another input event given before.
Please use instead 2 separate commands if possible.

```
e.g.    sample.enable  if  write&&int&&ab
                           ^          error
```

use instead

```
        sample.enable  if  write&&ab
        sample.enable  if  int&&ab
```

command or condition not realizable in current level

Due to hardware restrictions (HAC32) it isn't possible to use in this level the command mark.b or in the condition the input event BB.

This could be realized only in level 1 (c1) or 0 (c2).

countername in condition expected

Due to hardware restrictions (HAC32) it is necessary to include the defined counter event into the condition..

```
e.g.       eventcounter cnt0  200.
     LL0:
           counter.reset cnt0 if cnt0
           trigger.a          if read
                                   ^   error
           trigger.a          if read&&cnt0      // correct
```

unexpected command

Due to hardware restrictions (HAC32) it isn't possible to use in the same level the commands BREAK, CONTinue and GOTO simultaneously. Only one command of the three could be used in each level.

unexpected CONT

Due to hardware restrictions (HAC32) it's only possible to use one CONTinue command for each allowed trigger condition in each level.

```
Construction 1 (c1): in each level only    1 CONT command  possible
Construction 2 (c2): in level 0 at maximum 2 CONT commands possible
                     in level 1 at maximum 1 CONT command  possible
```

unexpected BREAK

Due to hardware restrictions (HAC32) it's only possible to use one BREAK command in the highest level.

not enough memory (for TriggerRAM Compiler)

Not enough system memory for the UPN-Buffer of the TriggerRAM Compiler.

not enough memory (for TriggerRAM Compiler)

Not enough system memory for the TriggerRAM Compiler.

global or local CONTinue ignored in last level

A given global or local CONTINUE command was been ignored in the last level of the triggerprogram.

not enough memory (for TriggerRAM Compiler)

Not enough system memory for codebuffer of the TriggerRAM Compiler.

not enough system memory free

Not enough system memory free for executing the command.

not enough system memory free

Not enough system memory free for executing the analyzer programming.

internal error : ANA_SetGlobalData.0

Not enough system memory for the global names.
Please call the manufacturer !

value out of range (0x0..0x10)

The value of the new deadtime is too small or too big.
Only values in the range of 0x0..0x10 are permitted.

value out of range (0x0..0x0f)

The value of the new unitnumber is too small or too big.
Only values in the range of 0x0..0x0f are permitted.

Error Messages

max. number of columns are reached

The max. number of columns for the list window of the performance analyzer was reached.

symbolname is too long

This symbolname exceeds the max. length.

symbol <*symbol*> not found in symboltable

no address or address-range

This definition is neither an address nor an addressrange.

bad address-range

The addressrange is not correct. The lower border is larger than the upper one.

area <*area*> has already been programmed at line <*line*>

Two areas are overlapping, check the definition of the ranges.

too many ranges

The number of ranges, covered by the analyzer is limited to 15, 31 or 63 areas. The SA120/HA120 units can cover 63 ranges in standard mode and 31 ranges when **ENTRY** is active. The ECC8 has 31 or 15 counters.

fatal error in performance analyzer

Error Messages

value out of range

Only values from 0. to 100. percent are allowed for the trigger delay.

analyzer is armed

invalid file type

Transmitter busy, clock o.k. ?

too many terminal windows

too many dump windows

WARNING: existing buffer may be insufficient for this baudrate

fast polling feature already used

no fast polling feature on this SCU16 need SCU32

no buffer memory available

channel name syntax error

no such channel

existing instruction overwritten

cannot combine BUS.C,D and communication analyzer

repeat or nesting overflow

closing bracket without open

pattern memory overflow, pattern too complex

no pattern defined

invalid file for binary pattern definition

WARNING: pattern generator uses channel with no CAT8 attached

WARNING: communication analyzer needs 50MHz sampling mode

WARNING: communication analyzer needs async channel D sampling

WARNING: pattern generator and communication analyzer use cannel D

WARNING: remove input/output probe attached to channel D

polarity is set twice

The polarity for the line was set two times.

```
e.g.   name.set a0 NMI- -
```

The first "-" will be interpreted as polarity too and not as a character from the new logical name if the switch name.polarity is OFF.

logical name expected

The logical name as a replacement of a physical name is forgotten.

```
e.g. wrong    name.set a0
     right    name.set a0 NMI
```

given name <*name*> exists twice

new logical name exist two times

The new logical name exists two times. The name exists already.

```
e.g.  1.  name.set A0 A1        or
      2.  name.set A0 NMI
          name.set A1 NMI
```

reserved name <*name*> used

A physical system name couldn't be used as a logical name.

```
e.g.  1.  name.set A0 A1        or
      2.  name.set A  B0        or
      3.  name.set C7 B
```

unexpected character in name

An unvalid character was detected in the name.

```
e.g. name.set A0 NMI/
```

unknown name used or syntax error

The given name doesn't exist. Only from the system predefined physical names or user-defined names can be used.

```
e.g.  name.set  0 a0 a0 -
```

fatal error in logical name definition, contact technical support (support@lauterbach.com)

fatal error in power probe

WARNING: no sync signal from SOC adapter

WARNING: SOC multiplexer fail

Error Messages

number in this context not allowed

A number is not allowed outside a name or a declaration expression.

":" expected

The closing '**:**' is missed in logical operator. Sometimes the rest of the expression was written in the next line.

```
e.g.    sample.enable if busd:A
                              ^       error position
        :N:full
```

unexpected character

unexpected EOF

'|' expected

second '**|**' character for **logical or** expected.

```
e.g. sample.enable if BUSD|!FULL
                          ^                 error position
     sample.enable if BUSD||!FULL        //  correct
```

'^' expected

second '**^**' character for **logical xor** expected.

```
e.g. sample.enable if BUSD^!FULL
                          ^                 error position
     sample.enable if BUSD^^!FULL        //  correct
```

'&' expected

second '**&**' character for **logical and** expected.

```
e.g. sample.enable if BUSD&!FULL
                          ^               error position
     sample.enable if BUS&&!FULL        //  correct
```

'/' expected

second '/' character for comment begin expected.

```
e.g. sample.enable   /7  enable record sampling
                      ^                              error position
     sample.enable   //  enable record sampling      correct
```

oldfashioned operator locked in current radix mode

In the current parser mode is the old syntax of operators and operands locked. Please switch mode to CLASSIC (**SETUP.RADIX CLASSIC**) or use new syntax.

old syntax operators: N: :A: :X: :O:

new syntax operators: ! && ^^ ||

```
e.g. sample.enable  if  N:TXD
                        ^  error
     sample.enable  if  !TXD
                        ^  ok
```

Please refer for details to chapter **parser changes** too !

not enough memory (for name table)

Not enough system memory for the expanded name table available.

too many names

Too many names used inside a timing analyzer progam.

declarations aren't allowed anymore

At this position in the timing analyzer program there is no declaration permitted. Declarations must stand before the first global instruction respectively the first level name (label).

```
e.g. "time  delay  10.ms"
     "s.on  if   delay"
     "event nr_int"
```

non declarable input variables not permitted here

Often there is here a command expected.

The non declarable input variables could be used only inside of conditions as input events (TRUE, BUSA, TRIGIN, …).

Often the command is written wrong and the given command name collides with a non declarable input event,

```
e.g. "busa" used instead of "bus.a".
e.g. "sample.on  if  cnt_event1"
     "busa"
      ^     errorposition
```

channel or data names not permitted here

Often there is here a command expected.
The (pre)defined channelnames or data event names could be used only inside of data event definitions.

```
Often the command is written wrong and the given command name collides with
a non declarable name,
e.g.  "bc"                // used instead of "bus.c".
e.g.  "s if cnt_event1"
      "bc"                // "bus.c" or "b.c" was ment, the dot was left out
```

EOF expected

unrecognized symbol;
superfluous (needles) symbols at a position where EOF is expected.

command expected

EOL expected

unrecognized symbol;
superfluous (needless) symbol at a position where EOL is expected
e.g. "out.a timer1" instead of "out.a c.i timer1"

unexpected EOL - command for command list expected

unexpected EOL - command for command list expected
normally command forgotten respectively command written in the next line

```
typical error:  1. line:  "s.on,  "
                2. line:  "c.on intno   if  entry&&int1"
```

unexpected EOL - condition for command list expected

unexpected EOL - condition for command list expected
normally condition forgotten respectively condition written in the next line

```
typical error:  1. line:  "s.on,  c.on intno   if  "
                2. line:  "entry&&int1"
```

separating BLANK expected

unexpected EOL - level name for goto expected

unexpected EOL - destination level name for goto command expected
normally name forgotten respectively name written in the next line

```
typical error:  1. line:  "goto  "
                2. line:  "start"
```

separating BLANK expected

label expected

keyword "IF" expected

unexpected EOL - condition expected

unexpected EOL - condition for command expected
normally condition forgotten respectively condition written in the next line

```
typical error:
  1. line:  "goto start if "
  2. line:  "write&&ab"
```

separating BLANK expected

keyword "IF" expected

separating BLANK expected

":" expected (for labelend)

unexpected symbol after label - ":" for labelend expected
normally labelend forgotten respectively labelend is written in the next lineor a command was written wrong

```
typical error:  1. line:  "level "
                2. line:  ":"
   or
                1. line:  "flagg.on"
```

label START must be first label

Due to hardware restrictions a label named START has to be the first written labelname.

unexpected EOL - ")" expected

unexpected EOL - ")" at the end of an condition is expected
normally ")" forgotten respectively ")" written in the next line

```
typical error:  1. line:  "goto start if (entry&&TRIGIN"
                2. line:  ")"
```

")" expected

no mode specification permitted

In the case of the given input event there is no mode specification possible or the input data event isn't defined.

```
typical error: "timecounter delay 10.ms"   or    "flags int"
               "s.e if delay.df"                  "s.e if int.df"
```

unexpected EOL - mode name for data or bus event expected

unexpected EOL - mode name for data or bus event expected
normally mode name forgotten respectively mode name written in the next linetypical error:

```
  1. line:  "goto start if dataev_reset."
  2. line:  "DF"
```

unexpected EOL - mode name for data event expected

unexpected EOL - mode name for data event expected
normally mode name forgotten respectively mode name written in the next linetypical error:

```
  1. line:  "goto start if ext.dataev_reset."
  2. line:  "DF"
```

mode name for data event expected

mode name is omitted or given mode name is not correct

```
e.g.:   "bus.a if ascii.&&V24RD ",  "bus.a if ascii.dff" or
        "bus.a if ascii. df",       "bus.a if ascii.d f"
```

mode name for data event expected

mode name is omitted or given mode name is not correct
```
e.g.:   "bus.a if word.ascii.&&V24RD ",  "bus.a if word.ascii.dff" or
        "bus.a if word.ascii. df",       "bus.a if word.ascii.d f"
```

mode name for bus event expected

mode name is omitted or given mode name is not correct
```
e.g.:   "s.e if busb.&&V24RD ",  "s.e if busb.fee" or
        "s.e if busb. fe",       "s.e if busb.f e"
```

unexpected data event prefix - currently locked

The used data event prefix is locked in the current used hardware.
EXT (X) and SOC (S) could be used only with the PowerProbe hardware.
Please use the keyword INTEGRATOR (I) instead.
```
e.g.:   "Sample.Enable IF I.A1
        "Sample.Enable IF Integrator.B5"
```

unexpected EOL - name of input event expected

unexpected EOL - name of input event at the end of an condition is expected. Normally the name is forgotten respectively was written in the next line.
```
typical error:  1. line:  "goto start if (entry&&"
                2. line:  "TRIGIN)"
```

name of input event expected

unexpected EOL - name for data event expected

unexpected EOL - name for data event expected
Normally the name is forgotten respectively was written in the next line.
```
typical error:  1. line:  "goto start if ext"
                2. line:  ".x1"
```

"." expected

"." before name for data channel, group or word event expected
```
e.g.  "goto start if ext. x1" instead of "goto start if ext.x1"
```

name for data event expected

name for data channel, group or word event expected
```
e.g.  "goto start if ext. x1"
      "goto start if ext.x 1"
      "goto start if ext.xx1" instead of "goto start if ext.x1"
```

unexpected data event prefix - currently locked

The used data event prefix is locked in the current used hardware.
I and INTEGRATOR could be used only with the PowerIntegrator hardware.
Please use the keyword EXT (X) or SOC (S) instead.

```
e.g.:    "Sample.Enable IF EXT.1||X.2
         "Sample.Enable IF SOC.1023||S.63"
```

command not allowed

Normally the command name is written wrong.

```
e.g.  "outt if datum1"
```

keyword for declaration required

ADDRESS selector definition not available for PowerProbe (PowerIntegrator only)

SELECTORRANGE selector definition not available for PowerProbe (PowerIntegrator only)

Analog probe must be plugged into connector A (PowerIntegrator only)

The event type SELECTORRANGE can be used only when an analog probe is plugged into the connector A of the PowerIntegrator modul.

ADDRESS selectors for actual hardware combination locked or CPU type not implemented

The PI address selector hardware for ADDRESS event usage inside the complex trigger programming, couldn't be used because the usage of PI hardware breakpoints is blocked from a different TRACE32 hardware e.g. PowerTrace (this kind of hardware breakpoints are prefered) or the PI hardware breakpoints aren't implemented for the actual used CPU type until now.
Please use SELECTORRANGE events instead.

ADDRESS and SELECTORRANGE events couldn't be used concurrently

ADDRESS and SELECTORRANGE events are using the identical hardware. Due to this, it isn't possible to use both types of events together in the same complex trigger program simultaneously.

unexpected EOL - address event name for address event expected

unexpected EOL - address event name for address event expected
Normally name forgotten respectively name written in the next line.

```
typical error:  1. line:  "address  "
                2. line:  "ab  up:1000--5500  "
```

separating BLANK expected

Frequently a wrong symbol is written instead of the address event name.

```
typical error :  "address , sp:1000"
                         ^
```

name of address event expected

unexpected EOL - address or addressrange expression expected

unexpected EOL - The value of the defined address event is expected.
Normally value forgotten respectively the value stands in the next line.

```
typical error:  1. line:  "address  alphabreak "
                2. line:  "  up:1000--5500     "
```

separating BLANK expected

Frequently a wrong symbol is written instead of the address event.

```
typical error :  "address alphabreak , sp:1000"
                                      ^
```

address or addressrange expected

The value of the defined address event is expected. Often only the access mode is left out.

```
e.g.       "address ab   0x10..0x1f||0x30"
instead of "address  ab p:0x10..0x1f||0x30"
```

address or addressrange expected

The value of the defined address event is expected. Often only the access mode is left out.

```
e.g. "address ab D:0x10 30s" instead of "address ab D:0x10 D:0x30"
```

unexpected EOL - data event name for data event or blank expected

unexpected EOL - Data event name for the data event or blank is expected.
Normally the name is forgotten respectively it stands in the next line.

```
typical error :  1. line:  "data  "
                 2. line:  " upper_char   a:'A'--'Z'"
```

unexpected EOL - name for selector event or blank expected

unexpected EOL - Name for the selector event or blank is expected.
Normally the name is forgotten respectively stands in the next line.

```
typical error :  1. line:  "selector  "
                 2. line:  " upper_char   word.b  'A'..'Z'"
```

separating BLANK expected

reserved name used

The use of names reserved from the system as names for data event is forbidden.

```
e.g. "data  TRUE  a:1"
```

name expected

unexpected EOL - data expression expected

unexpected EOL - The value of the defined data event is expected.
Frequently the expression is forgotten respectively stands in the next line.

```
typical error :  1. line:  "data  upper_char"
                 2. line:  "A:'A'--'Z'"
```

unexpected EOL - eXt, Group, Soc or Word expected

unexpected EOL - The defined data event needs a logical channel name (eXt.*, Soc.*), Word name (Word.*) or Group name (Group.*) to be assigned to it.
Frequently the name is forgotten respectively stands in the next line.
The available logical names could be displayed via **NAME.list** command.

```
typical error :  1. line:  "data  upper_char"
                 2. line:  "word.databus_B0 'A'--'Z'"
```

data expression expected

The user given expression has the wrong result type. Only expressions from the type integer (binary,hex,integer,ASCII), range, bit- / bytemask are possible.

```
e.g.:   "data ERROR  a:10.ms"     a time value isn't permitted
     or "data ERROR  d:d:10"      the address "d:10" isn't permitted
```

or

unexpected EOL - The value of the defined data event is expected.
Frequently the expression after the channel name is forgotten respectively stands in the next line.

```
typical error :  1. line:  "data  upper_char A:"
                 2. line:  "'A'--'Z'"
```

data expression expected

The user given expression has the wrong result type or is omitted. Only expressions from the type integer (binary,hex,integer,ASCII), range, bit- / bytemask are possible.

```
e.g.:  selector ERROR  ext.2 10.ms    a time value isn't permitted
 or    selector ERROR  ext.5 d:10     the address d:10 isn't permitted
 or   selector ERROR  ext.6 ext.nmi  the value for channel ext.6 was omitted
```

data event or channel name expected

eXt, Group, Soc or Word expected

The defined data event needs a logical channel name (eXt.*, Soc.*), Word name (Word.*) or Group name (Group.*) to be assigned to it.
Frequently the name is misspelled or partly written in the next line.
The available logical names could be displayed via **NAME.list** command.

```
typical error :  1. line:  "data  upper_char wor"
                 2. line:  "d.databus_B0 'A'--'Z'"
```

unknown data event or channel name

unexpected EOL - "." expected

unexpected EOL - The defined data event needs a "." after the logical channel name (eXt.*, Soc.*), Word name (Word.*) or Group name (Group.*) to assign the second name part to it.
Frequently the "." is forgotten respectively stands in the next line.

```
typical error :  1. line:  "selector  upper_char word"
                 2. line:  ".databus_B0 'A'--'Z'"
```

recursive name using not permitted

A name is used recursively in the same definition.

```
e.g.: "data  event1  event1 a:20"    wrong !
```

Only names (data events) which were defined before can be used.

```
e.g.: "data  event1  a:10"           correct !
      "data  event2  event1 a:20"
```

"." expected

The defined data event needs a "." after the logical channel name (eXt.*, Soc.*), Word name (Word.*) or Group name (Group.*) to assign the second name part to it.
Frequently the "." is forgotten respectively stands in the next line.

```
typical errors:  1. line:  "data  upper_char word "
                 2. line:  ".databus_B0 'A'--'Z'"
```

or

```
                 1. line:  "data  upper_char word databus_B0 'A'--'Z'"
```

negation not permitted

The negation of this expression is impossible because only values with one single fixed bit are allowed.

```
e.g.: "data int1   a:0xxxx0xx1!"              2 bits have fixed values !
      "data error  int1- a:20"

e.g.: "data int1    a:40"                     all 8 bits have fixed values !
      "data error  int1- a:20"

e.g.: "data int    a:0xxxxxxx1! a:0xxxxxx0x!"   2 values impossible !
      "data error  int- a:20"
```

unexpected EOL - second logical name expected

unexpected EOL - The defined data event needs after the "." of the logical channel name (eXt.*, Soc.*), Word name (Word.*) or Group name (Group.*) the second name part.
Frequently the name is forgotten respectively stands in the next line.
The available logical names could be displayed via **NAME.list** command.

```
typical error :  1. line:  "data  upper_char word."
                 2. line:  "databus_B0 'A'--'Z'"
```

unexpected EOL - ":" expected

unexpected EOL - ":" is expected. Frequently the ":" is forgotten respectively stands in the next line.

```
typical error :  1. line:  "data  upper_char  A"
                 2. line:  ":'A'--'Z'"
```

second logical name expected

The defined data event needs after "." of the logical channel name (eXt.*, Soc.*), Word name (Word.*) or Group name (Group.*) the second name part.
Frequently the name is misspelled or forgotten respectively stands in the next line.
The available logical names could be displayed via **NAME.list** command.

```
typical error :  "selector  upper_char  word.undefined_name 'A'--'Z'"
           or    "selector  upper_char  ext. 15  1"
           or    "selector  nmi         soc.1266 0"   // channel name 1266
unknown
```

unknown channel name - second logical name expected - wrong SOC MUX mode

The defined data event needs after "." of the logical channel name (Soc.*) the second name part.
Frequently the channel name is not selected via **Probe.SELect** respectively **Probe.TSYNC.SELect** command or the wrong soc multiplexer mode is configure **Probe.Mode**).
Or the name is misspelled or forgotten respectively stands in the next line.
The available logical names could be displayed via **NAME.list** command.

```
typical error :  "selector  nmi soc.nmi 0"      // channel name nmi unknown
```

too many channels used in word definition (>64)

The used Word name (Word.*) has more than 64 input channels.
The available logical names could be displayed via **NAME.list** command.

```
e.g.:  "selector  dsel0  W.WORD0 0xffff0000ffff0000"
```

":" expected

unexpected symbol - ":" is expected.

```
typical error :  "data  upper_char  A+:'A'--'Z'"
                                     ^   error pointer
```

unexpected EOL - data expression expected

unexpected EOL - The value of the defined data event is expected.
Frequently the expression after the channel name is forgotten respectively stands in the next line.

```
typical error :  1. line:  "data  upper_char A:"
                 2. line:  "'A'--'Z'"
```

channel name locked

The given channel is locked in the current timing analyzer frequency mode.

```
e.g. 100 MHz:    "data error c:20"          /* physical channel name */
     100 MHz:    "data error V24DATA:33"    /* system   channel name */
     200 MHz:    "data error B:33"
```

Group, Integrator or Word expected

The defined selector event needs a logical channel name (Integrator.*), Word name (Word.*) or Group name (Group.*) to be assigned to it.
Frequently the name is misspelled or partly written in the next line.
The available locigal names could be displayed via **NAME.list** command.

```
typical error :   1. line:  selector  upper5_char wor
                  2. line:  d.databus_B0  'A'..'Z'
```

The given channel is locked in the current timing analyzer frequency mode.

```
e.g. 100 MHz:    "data error c:20"          /* physical channel name */
     100 MHz:    "data error V24DATA:33"    /* system   channel name */
     200 MHz:    "data error B:33"
```

system data name locked

The given predefined system data event is locked in the current timing analyzer frequency mode. Is only in 50 MHz mode available.

```
e.g. 100 MHz:    "data error TXD"
     200 MHz:    "data error V24WR"
```

unexpected EOL - '.' expected

unexpected EOL - The defined selector event needs a "." after the logical channel name (I.*) to assign the mode to it.
Frequently the "." is forgotten respectively stands in the next line.

```
typical error :  1. line:  "SELECTOR  CSEL0  I.A0"
                 2. line:  ".HL "
```

user data name isn't usable

The given predefined user data event is locked in the current timing analyzer frequency mode because it is using locked physical channel.

"." expected

unexpected symbol - "." is expected.

```
typical error :  "selector  CSEL0  I.A0!HL
                                       ^   error pointer
```

unexpected EOL - channelmode expected

The defined selector event needs a mode after the logical channel name (I.A0.) to assign the mode to it.
Frequently the mode is forgotten respectively stands in the next line.

```
typical error :  1. line:  "SELECTOR  CSEL0  I.A0."
                 2. line:  "HL "
```

channelmode expected

The defined selector event needs a mode after the logical channel name (I.A0.) to assign the mode to it.
Often the mode name is written wrong or was omitted.

```
typical error :  SELECTOR  CSEL0  I.A0.EXT
```

unexpected EOL - username for selectorrange event expected

unexpected EOL - An of the user given name for time event is expected.
Normally name is forgotten respectively name is written in the next line.

```
typical error:  1. line:  "selectorrange  "
                2. line:  "too_high_voltage  0x1fc2--0x233c  "
```

separating BLANK expected

reserved name used

The use of names reserved from the system as names for selectorrange event is forbidden.

```
e.g. "selectorrange  TRUE  1000.--2000."
```

name for selectorrange event expected

unexpected EOL - integer or range expression expected

unexpected EOL - The value of the defined selectorrange event is expected.
Normally value forgotten respectively the value stands in the next line.

```
typical error:  1. line:  "selectorrange  too_high_voltage "
                2. line:  "  1000.--2000.     "
```

separating BLANK expected

Frequently a wrong symbol is written instead of the value.

```
typical error :  "selectorrange  too_high_voltage , 1000."
                                                   ^
```

numeric constant or range expected

The value of the defined selectorrange event is expected. Often an access mode is used for normal numeric ranges.

```
e.g. "selectorrange  too_high_voltage C:0x12--0x32" instead of
     "selectorrange  too_high_voltage 0x12--0x32"
```

numeric constant or range expected

The value of the defined selectorrange event is expected. Often an access mode is used for normal numeric ranges.

```
e.g. "selectorrange too_high_voltage 0x10 D:0x30 12s" instead of
     "selectorrange too_high_voltage 0x10 0x30"
```

unexpected EOL - username for time event expected

unexpected EOL - An of the user given name for time event is expected.
Normally name is forgotten respectively name is written in the next line.

```
typical error:  1. line:  "timecounter  "
                2. line:  "unvalid_time  1.ms--5.ms  "
```

separating BLANK expected

reserved name used

The use of names reserved from the system as names for time event is forbidden.
e.g. "timecounter TRUE 1.ns"

name expected

wrong expression type - time expression expected

unexpected EOL - username for event expected

unexpected EOL - An of the user given name for the counter event is expected.
Normally name is forgotten respectively name is written in the next line.

```
typical error:  1. line:  "eventcounter  "
                2. line:  "unvalid_path  0--5  "
```

separating BLANK expected

reserved name used

The use of names reserved from the system as names for counter event is forbidden.
e.g. "eventcounter TRUE 0ff"

name expected

wrong expression type - event expression expected

The given expression has not the type INTEGER or RANGE.

unexpected EOL - username for externcounter event expected

unexpected EOL - A user given name for the counter event is expected.
Normally the name is forgotten respectively the name is written in the next line.

```
typical error:  1. line:  externcounter
                2. line:  unvalid_path  0--5
```

separating BLANK expected

reserved name used

The use of names reserved from the system as names for counter event is forbidden.
e.g. externcounter TRUE 0xff

name expected

wrong expression type - event expression expected

The given expression has not the type INTEGER or RANGE.

unexpected EOL - username for flag expected

unexpected EOL - An of the user given name for the flag is expected.
Normally name is forgotten respectively name is written in the next line.

```
typical error :  1. line:  "flags  "
                 2. line:  "unvalid_path   "
       or        1. line:  "flag unvalid_path , "
                 2. line:  "     motor_on"
```

separating BLANK expected

reserved name used

The use of names reserved from the system as names for flag is forbidden.

```
e.g. "flags  TRUE "
```

flag name expected

unexpected symbol - An of the user given name for the flag is expected.
Normally a writting error occurred.

```
e.g. "flags , motor_on",  "flags .motor_on"
```

flag name expected

unexpected symbol - An of the user given name for the flag is expected.
Normally a writting error occurred.

```
e.g. "flags motor_on, n:LED_on"
      a.a.poaaaa..
```

channel already used before

Inside the selector definition channel names couldn't be used twice.
Sometimes this fact isn't obviously, when using WORD or GROUP names and single channel names together.
Please check the name definitions with the **NAME.list** command.

```
e.g. :  "selector  dsel0  EXT.0 1 EXT.1 0 EXT.0 0"
                                          ^  error position
```

loopvariable "?" not allowed in expression

value too big (max. <*bitnumber*> bit value)

The given counter value is hugher than the maximum 45bit value.

```
e.g. EVENTCOUNTER error_cnt  0x123456789012
                             ^              error position
```

value different from zero expected

only simple ranges are allowed

Only event ranges with 2 borders are possible.

data ranges are only allowed with byte borders

Due to hardware restrictions only ranges inside byte borders can be realized.
```
e.g.  DATA valid_values  word.d0_d7  0x01..0x1f
```

only byte range allowed

byte value expected in DATA declaration

value with max. <*bitnumber*> bit size expected

The actual given value exceeds the maximum possible value in bits.
```
e.g. SELECTOR nmi ext.1 0x2     // only values 0x0 or 0x1 expected
```

value with max. <*bitnumber*> bit size expected

The actual given value exceeds the maximum possible value in bits.
```
e.g. SELECTORRANGE too_high_value 0x1200--0x123456789
```

value with max. word size expected

value too big (max. 3 byte value)

byte value expected in DATA declaration for one byte channel

The user given value is no 1 byte value.
```
e.g. "data error  a:1234"
```

The channel A can only occupy byte values.

value with max. wordsize expected

value too big (max. 3 byte value)

value with max. 48 bit size expected

The actual given value exceeds the maximum possible counter value.

value too big (max. 4 byte value)

The actual given value exceeds the maximum possible data event value (32 bit).

unexpected EOL - subcommand for break command expected

unexpected EOL - A subcommand for the break command is expected.
Normally subcommand is forgotten respectively subcommand is written in the next line.
```
typical error :  1. line:  "break."
                 2. line:  "trace if cnt1"
```

invalid subcommand

subcommand is omitted or given subcommand is not correct
```
e.g.:   "break.  IF CNT1", "break.ttrace IF CNT1" or "break. trace IF CNT1"
```

unexpected EOL - "." expected

unexpected EOL - The introduction "." of a subcommand for the bus command is expected.
Normally "." is forgotten respectively is written in the next line.
```
typical error :  1. line:  "bus"
                 2. line:  ".a if cnt1"
          or     1. line:  "bus a if cnt1"
```

"." expected

unexpected EOL - subcommand expected

unexpected EOL - An subcommand for the bus command is expected.
Normally subcommand is forgotten respectively subcommand is written in the next line.
```
typical error :  1. line:  "bus."
                 2. line:  "a if cnt1"
```

invalid subcommand

subcommand is omitted or given subcommand is not correct
```
e.g.:   "bus.  IF CNT1", "bus.aa IF CNT1" or "bus. a IF CNT1"
```

unexpected EOL - counter subcommand expected

unexpected EOL - A subcommand of the counter command is expected.
Normally name is forgotten respectively name is written in the next line.
```
typical error :  1. line:  "counter."
                 2. line:  "on interrupt_events if INT1"
```

invalid subcommand

subcommand is omitted or given subcommand is not correct
```
e.g.:   "c. CNT1 IF INT", "c.onn CNT1 IF INT" or "c.o n CNT1 IF INT"
```

unexpected EOL - username for counter event expected

unexpected EOL - An of the user given name for the counter event (TIMECOUNTER, EVENTCOUNTER) is expected.
Normally name is forgotten respectively name is written in the next line.
```
typical error :  1. line:  "counter.on  "
                 2. line:  "interrupt_events"
```

separating BLANK expected

counter event name expected

unexpected CONTINUE

Due to hardware restrictions it isn't possible to use in the global level or in the highest level of the trigger program the command CONTINUE.

unexpected EOL - "." expected

unexpected EOL - A "." as a subcommand prefix of the flag command is expected.
Normally the "." is forgotten respectively is written in the next line.

```
typical error :  1. line:  "flag"
                 2. line:  ".on in_interrupt_fct if INT1"
```

"." expected

unexpected EOL - subcommand for flag command expected

unexpected EOL - An subcommand for the flag command is expected.
Normally subcommand is forgotten respectively subcommand is written in the next line.

```
typical error :  1. line:  "flag.  "
                 2. line:  "on interrupt_occurred   "
```

invalid subcommand

subcommand is omitted or given subcommand is not correct

```
e.g.:   "flag. in_int IF CNT1",    "flag.onn in_int IF CNT1" or
        "flag. on in_int IF CNT1", "flag.o n in_int IF CNT1"
```

unexpected EOL - username for flag expected

unexpected EOL - An of the user given name for the flag event is expected
Normally name is forgotten respectively name is written in the next line.

```
typical error :  1. line:  "flag.on  "
                 2. line:  "interrupt_occurred   "
```

separating BLANK expected

flag name expected

unexpected EOL - "." expected

unexpected EOL - The introduction "." of a subcommand for the out command is expected.
Normally "." is forgotten respectively is written in the next line.

```
typical error :  1. line:  "out"
                 2. line:  ".a if cnt1"
          or     1. line:  "out a if cnt1"
```

"." expected

unexpected EOL - subcommand for out command expected

unexpected EOL - An subcommand for the out command is expected.
Normally subcommand is forgotten respectively subcommand is written in the next line.

```
typical error :  1. line:  "out."
                 2. line:  "a if cnt1"
```

invalid subcommand

subcommand is omitted or given subcommand is not correct

```
e.g.:   "out.  IF CNT1", "out.aa IF CNT1" or "out. a IF CNT1"
```

unexpected EOL - subcommand for sample command expected

unexpected EOL - An subcommand for the sample command is expected.
Normally subcommand is forgotten respectively subcommand is written in the next line.

```
typical error :  1. line:  "sample."
                 2. line:  "on if cnt1"
```

invalid subcommand

subcommand is omitted or given subcommand is not correct

```
e.g.:   "sample.  IF CNT1", "s.oon IF CNT1" or "s. on IF CNT1"
```

unexpected "." - no subcommand expected

superfluous "." - the command TRIGGER or BREAK has no subcommands

"TRIGGER" or "BREAK" command not allowed

The commands TRIGGER and BREAK exclude each other.

A used command TRIGGER forbids in the whole trigger program the use of the command BREAK and vice versa. The command BREAK is equal to a command TRIGGER with a trigger delay of zero.

unexpected EOL - "." expected

unexpected EOL - The introduction "." of a subcommand for the trigger command is expected.
Normally "." is forgotten respectively is written in the next line.

```
typical error :  1. line:  "trigger"
                 2. line:  ".pattern if cnt1"
        or       1. line:  "trigger pattern if cnt1"
```

"." expected

unexpected EOL - subcommand for trigger command expected

unexpected EOL - An subcommand for the trigger command is expected.
Normally subcommand is forgotten respectively subcommand is written in the next line.

```
typical error :  1. line:  "trigger."
                 2. line:  "pattern if cnt1"
```

invalid subcommand

subcommand is omitted or given subcommand is not correct
e.g.: "trigger. IF CNT1", "trigger.pplus IF CNT1" or
"trigger. PLUS IF CNT1"

user data name <*name*> isn't usable

The given predefined user data event is locked in the current timing analyzer frequency mode because it is using locked physical channel.

level <*labelname*> doesn't exist

Not existing level labelname used in GOTO command.

data event <*name*> declared but not used

SELECTOR event <*name*> declared but not used in condition

TIMECOUNTER event <*name*> declared but not used in condition

TIMECOUNTER event <*name*> used but not set

TIMECOUNTER event <*name*> set, but not used in condition

EVENTCOUNTER condition <*name*> declared but not used in condition

EVENTCOUNTER condition <*name*> used, but never set

EVENTCOUNTER event <*name*> set, but not used in condition

FLAG <*name*> declared but not used

FLAG <*name*> used but not never set

FLAG <*name*> set, but not used

label START does not exist

Jump to the n o t existing level START.

too many counter used

At most a number of 2 universal counter can be used. They must be always used for time events and for counter events which are used in conditions.
The number of the used flags has to be subtracted from maximum number of counters.

too many counter used

At most a number of 3 universal counter can be used. They will be used for time events and for counter events.

too many flags used

At most a number of 2 flags can be used. Therefrom the number of the used counter has to be subtracted.

too many flags used

At most a number of 2 flags can be used.

too many input events - <*name*> does not fit

Too many global input events exist. The displayed flag name couldn't be fit in a free MUX.
The global instructions are "made" in every level (it seems they were done as local one; they have the same meaning as the local) and therefore they have the same restrictions as the local instructions.
At most 4 input events, which are using the internal multiplexers, are possible in each level.

too many input events - <*name*> does not fit in level <*levelname*>

Too many input events exist. The displayed flag name couldn't be fit in a free MUX.
The global instructions are "made" in every level (it seems they were done as local one; they have the same meaning as the local) and therefore they have the same restrictions as the local instructions.
At most 4 input events, which are using the internal multiplexers, are possible in each level.

EXTERNSYNCCOUNTER event <*name*> declared but not used in condition

EXTERNSYNCCOUNTER event <*name*> used but not set

EXTERNSYNCCOUNTER event <*name*> set, but not used in condition

SELECTORRANGE event <*name*> declared but not used in condition

too many SELECTORRANGE events used

At most a number of 8 SELECTORRANGE events can be used.

not declared name used

All names which could be freely created from the user have to be declared. The only exception are the labels. All data, time and counter events must be declared. Predefined system names are used for data events and that's the reason why no declaration is necessary.
Some of the predefined system names will be locked depending on the current selected analyzer speed.

too many data events used

At most 4 data events are possible in each level. Every global used data event reduces the number of the local (in a level) usable.

too many counter used

At most a number of 3 universal counter can be used. They will be used for time events and for counter events.

level name is declared twice

too many levels defined

At most 8 levels are possible. The actually used number of levels exceeds these limits.

too many levels defined

At most 4 levels are possible. The actually used number of levels exceeds these limits.

counter name not declared

counter <*name*> not declared

not enough memory (for internal table)

not enough memory (for internal table)

name <*name*> has multiple meaning

The name is used in two declarations or the name was used twice in the NAME.Set command.

```
e.g. 1. events NMI
        flags  NMI
     2. name.set a0 NMI
        flags  NMI
     3. name.set a1 a0 ==>  physical line a0 is logical name a0
                            physical line a1 is logical name a0 too
```

address event <*addr_name*> is multiple declared

The address event with the name addr_name has multiple declarations. This causes a special treatment of the declarations. All these declarations will be treated as one declaration (combined automatically).

```
e.g.      "address AB   sp:100         /hard"        are
          "address AB   v.range(flags) /hard"
equal to "address AB   sp:100 v.range(flags) /hard"
```

not enough memory (for internal table)

no levelname

The given name is no levelname.

level START not defined before the actual level, but must be the first level

Due to hardware restrictions a label named START has to be the first written labelname.

the line BUS<*busname*> is used before with an different mode

The given mode for the line BUS? isn't compatible with the mode used before. It's impossible to use a BUS? line simultaneously in different modes excepted in the case of '.State'.

```
e.g.:    's.on  if  busa.dh||busa.s'       ok !
         's.on  if  busa.dh||busa.df'      wrong
```

selected mode impossible

The given mode is different from the mode of an another dataevent which uses the same shared dataevent resource. Or the same dataevent is used with different modes in the same level.
Only 1 mode (excepted 'S') could be used in the whole program for the same physical dataevent.

too many dataevents used or the given mode is impossible in this program

At most 4 data events are possible in each level. Every global used data event reduces the number of the local (in a level) usable.
Only 1 mode (excepted 'S') could be used in the whole program for the same physical dataevent. In the present program configuration there are too many modes used. If not, then you can try to write the commands which are using the dataevent in their condition part, in a different order.

flag not declared

flag '<*name*>' not declared

flagname not unequivocal - name defined twice

All names can only be exist with one meaning and must be different from all keywords.

flagname '<*name*>' not unequivocal - name defined twice

All names can only be exist with one meaning and must be different from all keywords.

not enough memory (for internal table)

too many input events

Too many global input events exist.
The global instructions are "made" in every level (it seems they were done as local one; they have the same meaning as the local) and therefore they have the same restrictions as the local instructions.
At most 4 input events, which are using the internal multiplexers, are possible in each level.

counter <*name*> not declared

last level command overwritten

The current command overwrites the previous level command (CONTinue, GOTO) in the same line.

too many input events

Too many global input events exist.
The global instructions are "made" in every level (it seems they were done as local one; they have the same meaning as the local) and therefore they have the same restrictions as the local instructions.
At most 2 input events, which are using the internal multiplexers, are possible in each level.

too many input events

Too many global input events exist.
The global instructions are "made" in every level (it seems they were done as local one; they have the same meaning as the local) and therefore they have the same restrictions as the local instructions.
At most 4 input events, which are using the internal multiplexers, are possible in each level.

too many local input events

Too many local input events, using multiplexers exist. The number of input events in each level is the sum of the local used input events plus the global used input events.
At most 2 input events, which are using the internal multiplexers, are possible in each level.

too many local input events

Too many local input events, using multiplexers exist. The number of input events in each level is the sum of the local used input events plus the global used input events.
At most 4 input events, which are using the internal multiplexers, are possible in each level.

breakpoint type <breakpointname> not set and for this always FALSE

The given breakpoint inside a condition isn't defined as an address event in the trigger program nor set outside the trigger program with the Break.Set command.

```
e.g.    address ab p:1000
        sample.enable  if  bb
                           ^        warning position
```

not enough memory

Not enough system memory for internal table.

not enough memory

Not enough system memory for internal command table.

not enough memory (for TA32||PP||PI RAM Compiler)

Not enough system memory for the UPN buffer of the TA/PP/PI RAM compiler.

not enough memory (for TriggerRAM Compiler)

Not enough system memory for the TriggerRAM Compiler.

global or local CONTinue ignored in last level

A given global or local CONTINUE command was been ignored in the last level of the triggerprogram.

not enough memory (for TA32||PP||PI compiler)

Not enough system memory for codebuffer of the Timing Analyzer compiler.

not enough system memory free

Not enough system memory free for executing the command.

BREAK command changed trigger delay to zero

internal error : TA_SetGlobD.int

Not enough system memory for the global names.
Please call the manufacturer !

not enough memory

Not enough system memory for internal table.

internal error: unexpected empty pointer in PPTA_PI_COMBINE_AND_WRITE_DSELS

The given several values aren't realizable: <*channelgroup*><*selector*>

The wished combination of 2 or more values couldn't be realized with one selector due to hardware restrictions.
Please split the selector into two or try to rearange the input channel order (ranges are only possible in one channel group).

```
e.g. NAME.SET WORD.0 X.0 X.1 X.2 X.3  X.8 X.9 X.10 X.11
     SELECTOR dsel0  WORD.0 0x10 0x02
```

deadtime value out of range (0..16)

only the first 102 input channel can be used for AMUX definition

baseaddress value out of range (0x0..0xffffffff)

base input channel already in use

for definition of BASE.<cs_no> no free mux line left (CSMUXRAM)

no AMUX/BMUX settings were done before

not enough memory (for PI supportpackage setting)

Not enough system memory for the support package programming of PI.

CPU specific PI hardware breakpoints aren't usable or implemented

The PI address selector hardware for ADDRESS event usage inside the complex trigger programming, couldn't be used because the usage of PI hardware breakpoints is blocked from a different TRACE32 hardware e.g. PowerTrace (this kind of hardware breakpoints are prefered) or the PI hardware breakpoints aren't implemented for the actual used CPU type until now.
Please use SELECTORRANGE events instead.

internal error : <*error*>

Please contact the manufacturer !

Error Messages

Error (<*code*>) in selftest of programmer

The error may occur, if a chip is in the socket during power-up or if there is a fatal hardware fault. During operation this error can occur if the electronic fuse is activated.The programmer cannot operate after this error message.

invalid address, must be byte aligned

negative address not allowed

pinnumber of testvectors and chip do not match

load list full, no more load jobs possible

no such chip

no such job

breaked

joblist full, no more jobs possible

The size of the joblist can be configured with the command **Option.Job**.

invalid data width, matches not with chip

data width double redefined

wrong bit range

invalid address for chip

invalid address for buffer

no jobs defined

error accessing virtual memory

error in XPRO database

no signature

double signature

too many selection items

item unknown

Size, manufacturer, package or type of chip must be entered here.

item here not allowed, use '*'

not programmable with current hardware

A higher level of the programmer, or additional adapters are required.

no such device

no data in buffer

window too large

UES signature too long

programmer busy

To break use the command **BREAK**.

chip in socked

warning: size of chip (<*size*>) and file (<*size*>) not equal

warning: fusenumber entry of file and chip do not match

warning: pinnumber entry of file and chip do not match

warning: device code of file and chip do not match

warning: device code and architecture of file do not match with chip

warning: device code of file not identified

warning: file is converted to chip format

file cannot be converted

device cannot be converted

device conversion error

internal error: VALSIZE (contact technical support (support@lauterbach.com))

internal error: ALGSIZE (contact technical support (support@lauterbach.com))

internal error: MAPSIZE (contact technical support (support@lauterbach.com))

internal error: PINSIZE (contact technical support (support@lauterbach.com))

internal error: CONVSIZE (contact technical support (support@lauterbach.com))

Error Messages

single bits not allowed

no such pod name