

Message Class 0001 - RPC C Runtime

This page describes messages with message class 0001. These messages have the format:

0001nnnn

where 0001 is the message class and

nnnn is the message number in the range 0000 - 9999.

This message class is returned if an error has occurred in the RPC C runtime layer.

Overview of Messages

00010001	00010003	00010004	00010005	00010006	00010007
00010008	00010009	00010010	00010011	00010018	00010019
00010028	00010032	00010034	00010036	00010037	00010038
00010039	00010045	00010046	00010052	00010053	00010054
00010055	00010056	00010058	00010060	00010061	00010063
00010064	00010065	00010066	00010067	00010068	00010069
00010070	00010071	00010072	00010075	00010076	00010077
00010078	00010079	00010080	00010081	00010082	00010083
00010084	00010085	00010086	00010087	00010088	00010089
00010090	00010091	00010094	00010095	00010096	00010097
00010100	00010101	00010102	00010103	00010104	00010105
00010106	00010107	00010108	00010109	00010110	00010200
00010201	00010202	00010203	00010204	00010205	00010206
00010207	00010208	00010209	00010210	00010211	00010212
00010213	00010215	00010216	00010217	00010218	00010219
00010220					

00010001 No Communication

Explanation Denotes an inconsistency within the EntireX RPC Runtime.

Action Contact Software AG support.

00010003 Unknown medium

Explanation The EntireX RPC Runtime detected an invalid server address.

Action When using the C API *erx.h*:
The ERX_SERVER_ADDRESS structure of the EntireX RPC Runtime is not correct or was not correctly initialized. The medium must be ERX_TM_BROKER, ERX_TM_BROKER_LIBRARY or ERX_TM_CONNECTION.

00010004 Invalid control block

Explanation The EntireX RPC Runtime API was called with an invalid connection or `CallId`.

Action When using the C API *erx.h*:
Check program and correct error.

00010005 Out of memory

Explanation The operating system could not satisfy a memory request.

Action Increase your memory resources and retry.

00010006 No more registrations

Explanation An `ERXRegister` API call can register a maximum of 256 threads at a time with the EntireX RPC Runtime within one process.

Action When using the C API *erx.h*:
Check program and correct error.

00010007 Already registered

Explanation An `ERXRegister` API call is attempting to register a thread with the EntireX RPC Runtime twice.

Action When using the C API *erx.h*: It is not necessary to register a thread multiple times. However, it is required to register a thread once. We recommend having the number of `ERXRegister`/`ERXUnregister` calls occurring in pairs. If you insist on registering a thread twice, ignore this error. Check program and correct error.

00010008 Not registered in ERX Runtime

Explanation Before a thread/process can use an EntireX RPC Runtime API function, the thread/process must be registered with the EntireX RPC `ERXRegister` API call.

Action

- When using the C API *erx.h*:
Check program and correct error.
- When using the DCOM Wrapper:
Contact Software AG support.

00010009 Parameter error. Location of wrong parameter is n

Explanation An EntireX RPC Runtime API function was called with an invalid address or inconsistent parameter. The number n denotes the position of the inconsistent parameter within the API.

Action When using the C API *erx.h*:
Check program and correct error.

00010010 Control block not found

Explanation An EntireX RPC Runtime API function with corrupt memory was called. The internal data structures were not found.

Action

- When using the C API *erx.h*:
Check program and correct error.
- When using the DCOM Wrapper:
Contact Software AG support.

00010011 Pending call

Explanation During a conversational Remote Procedure Call there was a failure and another EntireX RPC Runtime `ERXCall` API call was issued.

Action When using the C API *erx.h*:
Close the Connection with the EntireX `ERXDisconnect` API function call.

00010018 Invalid memory

Explanation One of the following occurred:

1. The memory area supplied for the EntireX RPC Runtime API is invalid. It could not be read from or written to.
2. An `ERX_VDATA` handle is corrupt.

Action

- When using the C API *erx.h*:
Check program and correct error.
- When using the DCOM Wrapper:
Contact Software AG support.

00010019 Invalid Call Id

Explanation The CallId for the EntireX RPC Runtime API function is invalid or corrupt.

Action

- When using the C API *erx.h*:
Check program and correct error.
- When using the DCOM Wrapper:
Contact Software AG support.

00010028 Illegal type

Explanation The EntireX RPC Runtime API received an invalid parameter. There is an inconsistency between the template files, the header file *erx.h* or the EntireX RPC Runtime. The type of parameter is not supported.

Action

- When using the C API *erx.h*:
Re-generate the client proxy or server stubs with the correct template file *client.tpl* and *server.tpl* and compile and link it with the header file *erx.h* from the same EntireX RPC version.
- When using the DCOM Wrapper:
Re-generate the DCOM server, using the correct template files.

00010032 Too many indices

Explanation The EntireX RPC Runtime API call received an invalid parameter. There is an inconsistency between the template files, the header file *erx.h* or the EntireX RPC Runtime.

Action The EntireX RPC Runtime is binary backward compatible. It must have the same or a higher EntireX version as the template and header files used.

- When using the C API *erx.h*:
Re-generate the client proxy or server stubs with the correct template file *client.tpl* and *server.tpl* and compile and link it with the header file *erx.h* from the same EntireX RPC version.
- When using the DCOM Wrapper:
Re-generate the DCOM server, using the correct template files.

00010034 No pending call

Explanation An EntireX RPC Runtime API `ERXWait` call was issued and no RPC request was pending.

Action Check program and correct error.

00010036 Parameters values not ascending

Explanation The EntireX RPC Runtime API call received an invalid parameter. There is an inconsistency between the template files, the header file *erx.h* or the EntireX RPC Runtime. One of the following applies:

- The size of the first dimension of an array parameter must not be less than the size of the second dimension.
- The size of the second dimension of an array parameter must not be less than the size of the third dimension.

Action The EntireX RPC Runtime is binary backward compatible. It must have the same or a higher EntireX version as the template and header files used.

- When using the C API *erx.h*:

Re-generate the client proxy or server stubs with the correct template file *client.tpl* and *server.tpl* and compile and link it with the header file *erx.h* from the same EntireX RPC version.

00010037 Invalid timeout

Explanation The timeout value must not be zero.

Action ● When using the C API *erx.h*:

The `ERX_SERVER_ADDRESS` structure of the EntireX RPC Runtime is not correct or was not correctly initialized. Check program and correct error.

- When using the DCOM Wrapper:

Provide a valid timeout for the property `TimeOut`.

00010038 Version not supported

Explanation The EntireX RPC Runtime API `ERXRegister` call was issued with an invalid version.

Action The EntireX RPC Runtime is binary backward compatible. It must have the same or a higher EntireX version as the template and header files used.

- When using the C API *erx.h*:

Use the appropriate `ERX_Vnnn` macro defined in the *erx.h* header file.
Re-generate the client proxy or server stubs with the correct template file *client.tpl* and *server.tpl* and compile and link it with the header file *erx.h* from the same EntireX RPC version.
- When using the DCOM Wrapper:

Re-generate the DCOM server, using the correct template files.

00010039 Cyclic parameter

Explanation The EntireX RPC Runtime API call received an invalid parameter. There is an inconsistency between the template files, the header file *erx.h* or the EntireX RPC Runtime. The offending parameter (or one of its parents) is its own parent. The parameter definition is cyclic.

Action The EntireX RPC Runtime is binary backward compatible. It must have the same or a higher EntireX version as the template and header files used.

- When using the C API *erx.h*:

Re-generate the client proxy or server stubs with the correct template file *client.tpl* and *server.tpl* and compile and link it with the header file *erx.h* from the same EntireX RPC version.
- When using the DCOM Wrapper:

Re-generate the DCOM server, using the correct template files.

00010045 AT EXIT error

Explanation An error occurred while establishing the `atexit` function.

Action Contact Software AG support.

00010046 Initializing trace failed

Explanation Tracing for the EntireX RPC Runtime could not be switched on. Either no trace destination had been entered, or the trace file could not be opened.

Action Contact your System Administrator.

00010052 Library missing

Explanation No Library name was given.

Action ● When using the C API *erx.h*:

The ERX_SERVER_ADDRESS structure of the EntireX RPC Runtime is not correct or was not correctly initialized. When using the ERX_TM_BROKER_LIBRARY as the medium, szLibraryName must also be given. Check program and correct error.

● When using the DCOM Wrapper:

Provide a valid library for the property Library.

00010053 User ID missing

Explanation No User ID was entered.

Action ● When using the C API *erx.h*:

The ERX_CLIENT_IDENTIFICATION structure of the EntireX RPC Runtime is not correct or was not correctly initialized. Check program and correct error.

● When using the DCOM Wrapper:

Provide a valid user ID for the property UserID.

00010054 Password missing

Explanation No Password was entered.

Action 1. When using the C API *erx.h*:

The ERX_CLIENT_IDENTIFICATION structure of the EntireX RPC Runtime is not correct or was not correctly initialized. Check program and correct error.

2. When using the DCOM Wrapper:

Provide a valid password for the property PassWord.

00010055 BCD overflow

Explanation This is an internal error.

Action Contact Software AG support.

00010056 BCD division by zero

Explanation This is an internal error.

Action Contact Software AG support.

00010058 Invalid direction

Explanation The EntireX RPC Runtime API call received an invalid parameter. The direction specification must be one of the following: ERX_IN_PARM, ERX_OUT_PARM or ERX_INOUT_PARM.

Action The EntireX RPC Runtime is binary backward compatible. It must have the same or a higher EntireX version as the template and header files used.

- When using the C API *erx.h*: Re-generate the Client proxy or Server stubs with the correct template file *client.tpl* and *server.tpl* and compile and link it with the header file *erx.h* of the same EntireX RPC version.
- When using the DCOM Wrapper:

Re-generate the DCOM server, using the correct template files.

00010060 Response <response> from(Command/Info Service)

Explanation A response was retrieved from the broker's Command/Info Service interface while an EntireX RPC API function `ERXTerminateServer` was executed.

Action See *Broker Command and Information Services Error Codes*.

00010061 Server not found(Command/Info Service)

Explanation This is an informational message of the EntireX RPC Runtime.

Action Contact Software AG support.

00010063 Invalid Parameter Definition Version

Explanation The EntireX RPC parameter definition is not consistent to the EntireX version registered with the EntireX RPC API `ERXRegister` function.

There is an inconsistency between the template files, the header file *erx.h* or the EntireX RPC Runtime.

Action

- When using the C API *erx.h*:

For all versions prior to `ERX_V530`, `ERX_PARAMETER_DEFINITION` (defined in *erx.h* of related version) must be used. For `ERX_V530` and above `ERX_PARAMETER_DEFINITION_V2` (defined in *erx.h* of related version) must be used. Re-generate the client proxy or server stubs with the correct template file *client.tpl* and *server.tpl* and compile and link it with the header file *erx.h* of the same EntireX RPC version.

- When using the DCOM Wrapper:

Re-generate the DCOM server, using the correct template files.

00010064 Internal error

Explanation This is an internal, informational message of the EntireX RPC Runtime.

Action Contact Software AG support.

00010065 No more VData Instances to free

Explanation This is an internal, informational message of the EntireX RPC Server.

Action Contact Software AG support.

00010066 Cache item not found

Explanation This is an internal, informational message of the EntireX RPC Runtime.

Action Contact Software AG support.

00010067 Success pending call

Explanation This can be a result of a `ERX_CC_PEEK` command of the EntireX RPC Runtime API function `ERXControl`.

Action Program dependent.

00010068 Not yet implemented

Explanation This is an internal, informational message of the EntireX RPC Runtime.

Action Contact Software AG support.

00010069 Invalid string buffer

Explanation While examining the string buffer, an error occurred within the EntireX RPC Runtime. The partner did not adhere to the metastructure <keyword>=<length>,<value>.

Action Contact Software AG support.

00010070 Parse error

Explanation This is an internal, informational message of the EntireX RPC Runtime.

Action Contact Software AG support.

00010071 Server replied with different compression

Explanation Your EntireX/Natural RPC server does not reply correctly to the EntireX RPC request.

Action Contact Software AG support.

00010072 Return Code <return-code> <cb-return-code>from Broker Stub

Explanation The EntireX Broker Stub replied with the specified return codes, where <return-code> is the stub's function return code and <cb-return-code> is from the ACI field return code.

Action Check your configuration and correct the error.

00010075 No RPC protocol version received by INQUIRE Call

Explanation The EntireX/Natural RPC server does not reply correctly to the INQUIRE call.

Action Contact Software AG support.

00010076 Server does not support RPC protocol version needed for the request

Explanation The request issued by the RPC client cannot be processed by the RPC server because the RPC server or - if it is being used - the broker conversion approach SAGTRPC does not support the required functionality.

Action If the broker conversion approach SAGTRPC is being used, the cause of the problem can be SAGTRPC or the RPC server. Check the version of the broker and the RPC server in use. Upgrade the broker and the RPC server to the latest version available.

 If the broker conversion approach SAGTRPC is not being used, the cause of the problem is the RPC server. Check the version of the RPC server in use. Upgrade the RPC server to the latest version available.

00010077 Broker API Version 2 needed

Explanation Your EntireX Broker middleware (stub or kernel) does not support the Broker API version needed.

Action If possible, upgrade your EntireX Broker Middleware (Stub and Kernel) at least to ETB211.

00010078 Parameter maximum length or array bounds overflow

Explanation There was an overflow of an unbounded array. The maximum array boundaries are determined in the Software AG IDL file.

Action Check the program and correct the error.

00010079 Invalid unbounded array

Explanation Accessing an unbounded array has caused an error. This can happen during marshalling/unmarshalling within the EntireX RPC Runtime or when accessing the `erxArray...` functions directly.

Action Either:

1. The unbounded array was destroyed by your program because of memory overwrites. Correct your program.

 or

2. An internal error occurred in the EntireX RPC Runtime. Contact Software AG support.

00010080 No more unbounded arrays to free

Explanation This is an internal, informational message of the EntireX RPC Server.

Action Contact Software AG support.

00010081 Indices not correct

Explanation The `ERX_PARAMETER_DEFINITION_Vx` contains incorrect data in the `usAttributes` and `usType` fields.

Action Re-generate, compile and link your client stub.

00010082 Unbounded array does not match the IDL file

Explanation The data type and attributes of the unbounded array built with `erxArrayAlloc` do not match the data type, length, dimensions or attributes determined in the Software AG IDL file.

Action Re-generate your client stub or create the unbounded array with the correct type, length, dimensions and attributes using `erxArrayAlloc`.

00010083 Parameter not within value range

Explanation In the Software AG IDL file value ranges are defined for data types. The value you are trying to send is outside of the defined range. For example, it is not possible to send

- a negative value with data type NU or PU
- a value greater than 127 and less than -128 with I1
- a value greater than 32767 and less than -32768 with I2
- a value greater than 2147483647 and less than -2147483648 with I4.

Action Check the program and correct the error.

00010084 Unbounded array indices out of bounds

Explanation The indices you provided to access the unbounded array are out of bounds.

Action Check the program and correct the error.

00010085 Invalid data for unbounded array

Explanation The data provided for the unbounded array function does not conform to the rules of the data type. For example, the data provided for a data type P is not packed.

Action Check the program and correct the error.

00010086 Invalid preserve flag for unbounded array

Explanation The parameter preserve flag provided for the unbounded array function is invalid.

Action Check the program and correct the error.

00010087 Invalid dimension for unbounded array

Explanation The parameter dimension provided for the unbounded array function is invalid.

Action Check the program and correct the error.

00010088 Unbounded array parameter error

Explanation The parameter provided for the unbounded array function is invalid.

Action Check the program and correct the error.

00010089 Stub parameter definition error

Explanation The parameter definition within the client stub is incorrect.

Action Re-generate your client stub with the IDL Compiler or EntireX DCOM Wrapper.

00010090 SSL parameter initialization failed

Explanation Initialization failed because EntireX Broker stub does not support SSL.

Action Use a stub which supports SSL or configure a transport method supported by the stub used.

00010091 Broker API Version 6 needed

Explanation Your EntireX Broker middleware (Stub or Kernel) does not support the Broker API version needed.

Action If possible, upgrade your EntireX Broker middleware (Stub and Kernel).

00010094 Broker Stub could not be loaded

Explanation The EntireX Broker stub could not be found in the environment.

Action Try again to install EntireX.

00010095 Cannot open or read File=<file name>

Explanation An invalid configuration file name or a wrong file type was used to start the RPC server.

Action Restart the server with correct configuration file.

00010096 Error code for reply error missing

Explanation This is an internal error of the RPC runtime and RPC server.

Action Contact Software AG support.

00010097 Password change not supported by RPC server

Explanation The RPC server you are using does not support alteration of the RPC password. Altering RPC passwords is supported by Natural RPC Server only. EntireX RPC servers do not support alteration of RPC passwords.

Action Contact Software AG support to check whether a newer version of the Natural RPC Server may support the feature.

00010100 ICONV Open error for Unicode conversions

Explanation The RPC Runtime uses the iconv interface for converting Unicode encodings under UNIX. The iconv interface returned an error (-1) during open of the conversion descriptor. Either the source or target Unicode encoding are not supported.

Action The Unicode encoding UTF-16BE is always needed. Depending on the used Unicode encoding of your environment, one of the following additional encodings are required: UTF-32BE UTF-32LE or UTF-16LE. See your UNIX documentation for more information on the iconv interface, especially the iconv open call. Check if the above encodings are available in your UNIX environment.

00010101 ICONV target buffer for Unicode conversions too small

Explanation The RPC Runtime uses the iconv interface for converting Unicode encodings under UNIX. The iconv interface returned errno E2BIG. This error is an internal error of the RPC Runtime.

Action See your UNIX documentation for more information on the iconv interface and error E2BIG. Contact Software AG support.

00010102 ICONV Invalid descriptor

Explanation The RPC Runtime uses the iconv interface for converting Unicode encodings under UNIX. The iconv interface returned errno EBADF. This error is an internal error of the RPC Runtime.

Action See your UNIX documentation for more information on the iconv interface and error EBADF. Contact Software AG support.

00010103 ICONV Invalid Character or Shift Sequence

Explanation The RPC Runtime uses the iconv interface for converting Unicode encodings under UNIX. The iconv interface returned errno EILSEQ. This is a user error.

Action Check your data if valid Unicode characters being sent/received. See your UNIX documentation for more information on the iconv interface and error EILSEQ.

00010104 ICONV Incomplete Character or Shift Sequence

Explanation The RPC Runtime uses the iconv interface for converting Unicode encodings under UNIX. The iconv interface returned errno EINVAL. This is a user error.

Action Check your data to see whether valid Unicode characters are being sent/received. See your UNIX documentation for more information on the iconv interface and error EINVAL.

00010105 ICONV Failed

Explanation The RPC Runtime uses the iconv interface for converting Unicode encodings under UNIX. The iconv interface returned an undefined error.

Action See your UNIX documentation for more information on the iconv interface. Contact Software AG support.

00010106 Illegal parameter direction

Explanation Reliable RPC allows only IN parameters.

Action Remove any other OUT and INOUT parameters from this message.

00010107 Invalid packet type for Reliable RPC

Explanation The packet type shipped with the RPC protocol is not valid for the use of Reliable RPC.

Action Contact Software AG support.

00010108 Inconsistent Reliable RPC message

Explanation The string buffer item "REL" is missing or invalid.

Action Contact Software AG support.

00010109 Reserved for internal use

00010110 Reserved for internal use

**00010200 Parameter Error: data not unsigned numeric. <direction> of IDL field
[<parameter-name>] (<type-length>), location=<location>**

Explanation An application parameter cannot be processed because its content does not match the expected format unsigned numeric (NU). See *IDL Data Types*.

The message provides additional information such as <direction>, <parameter-name>, <type-length> and <location>, where

- <direction> is either 'Request' if the error was detected before the RPC server was executed, or 'Reply' for the answer, i.e. if the RPC server was already executed.
- <parameter-name> is available for target programming language C only
- <type-length> matches the IDL data type; see *IDL Data Types*
- <location> contains the hierarchy of IDL nodes, separated by a colon. If an IDL node is an array, its indexes are given in parentheses.

Example: Reply of IDL field (NU14.0), location=3.2(I=5).1.

The IDL below was used for the RPC call. The location is interpreted as follows: Third field on level 1 is 'G-parameter-3-on-level-1'; then second parameter on level 2 is the array 'G-parameter-32-on-level-2' and (I=5) means the fifth occurrence; then the first field on level 3 which is 'G-parameter-322-on-level-3' produced the error.

```

. . .
define data parameter
1 G-parameter-1-on-level-1      (A10) In
1 G-parameter-2-on-level-1      In
1 G-parameter-3-on-level-1      Out
  2 G-parameter-31-on-level-2
    3 G-parameter-311-on-level-3 (A5)
    3 G-parameter-312-on-level-3 (N8)
    3 G-parameter-313-on-level-3 (NU4)
  2 G-parameter-32-on-level-2   (/25)
    3 G-parameter-322-on-level-3 (NU14)
    3 G-parameter-321-on-level-3 (A5)
1 G-parameter-4-on-level-1      (A64)
. . .
end-define

```

Action Make sure the IDL file, generated client interface objects and your client application are consistent. Check your client application for parameter integrity.

**00010201 Parameter Error: data not signed numeric. <direction> of IDL field
[<parameter-name>] (<type-length>), location=<location>**

Explanation An application parameter cannot be processed because its content does not match the expected format signed numeric (N). See *IDL Data Types*.

The additional information <direction>, <parameter-name>, <type-length> and <location> is explained under error message 00010200.

Action Make sure the IDL file, generated client interface objects and your client application are consistent. Check your client application for parameter integrity.

**00010202 Parameter Error: data not numeric. <direction> of IDL field [<parameter-name>]
(<type-length>), location=<location>**

Explanation An application parameter cannot be processed because its content does not match the expected format unsigned numeric (NU) or signed numeric (N). See *IDL Data Types*.

The additional information <direction>, <parameter-name>, <type-length> and <location> is explained under error message 00010200.

Action Make sure the IDL file, generated client interface objects and your client application are consistent. Check your client application for parameter integrity.

**00010203 Parameter Error: data not unsigned packed. <direction> of IDL field
[<parameter-name>] (<type-length>), location=<location>**

Explanation An application parameter cannot be processed because its content does not match the expected format unsigned packed (PU). See *IDL Data Types*.

The additional information <direction>, <parameter-name>, <type-length> and <location> is explained under error message 00010200.

Action Make sure the IDL file, generated client interface objects and your client application are consistent. Check your client application for parameter integrity.

**00010204 Parameter Error: data not signed packed. <direction> of IDL field
[<parameter-name>] (<type-length>), location=<location>**

Explanation An application parameter cannot be processed because its content does not match the expected format signed packed (P). See *IDL Data Types*.

The additional information <direction>, <parameter-name>, <type-length> and <location> is explained under error message 00010200.

Action Make sure the IDL file, generated client interface objects and your client application are consistent. Check your client application for parameter integrity.

00010205 Parameter Error: data not packed. <direction> of IDL field [<parameter-name>] (<type-length>), location=<location>

Explanation An application parameter cannot be processed because its content does not match the expected format unsigned packed (PU) or signed packed (P). See *IDL Data Types*.

The additional information <direction>, <parameter-name>, <type-length> and <location> is explained under error message 00010200.

Action Make sure the IDL file, generated client interface objects and your client application are consistent. Check your client application for parameter integrity.

00010206 Parameter Error: I1 < MIN. <direction> of IDL field [<parameter-name>] (<type-length>), location=<location>

Explanation An application parameter of IDL type I1 cannot be processed because its content is less than the allowed minimum (-128) for I1. See *IDL Data Types*.

The additional information <direction>, <parameter-name>, <type-length> and <location> is explained under error message 00010200.

Action Make sure the IDL file, generated client interface objects and your client application are consistent. Check your client application for parameter integrity.

00010207 Parameter Error: I1 > MAX. <direction> of IDL field [<parameter-name>] (<type-length>), location=<location>

Explanation An application parameter of IDL type I1 cannot be processed because its content is greater than the allowed maximum (+127) for I1. See *IDL Data Types*.

The additional information <direction>, <parameter-name>, <type-length> and <location> is explained under error message 00010200.

Action Make sure the IDL file, generated client interface objects and your client application are consistent. Check your client application for parameter integrity.

00010208 Parameter Error: I2 < MIN. <direction> of IDL field [<parameter-name>] (<type-length>), location=<location>

Explanation An application parameter of IDL type I2 cannot be processed because its content is less than the allowed minimum (-32768) for I2. See *IDL Data Types*.

The additional information <direction>, <parameter-name>, <type-length> and <location> is explained under error message 00010200.

Action Make sure the IDL file, generated client interface objects and your client application are consistent. Check your client application for parameter integrity.

00010209 Parameter Error: I2 > MAX. <direction> of IDL field [<parameter-name>] (<type-length>), location=<location>

Explanation An application parameter of IDL type I2 cannot be processed because its content is greater than the allowed maximum (+32767) for I2. See *IDL Data Types*.

The additional information <direction>, <parameter-name>, <type-length> and <location> is explained under error message 00010200.

Action Make sure the IDL file, generated client interface objects and your client application are consistent. Check your client application for parameter integrity.

00010210 Parameter Error: I4 < MIN. <direction> of IDL field [<parameter-name>] (<type-length>), location=<location>

Explanation An application parameter of IDL type I4 cannot be processed because its content is less than the allowed minimum (-2147483648) for I4. See *IDL Data Types*.

The additional information <direction>, <parameter-name>, <type-length> and <location> is explained under error message 00010200.

Action Make sure the IDL file, generated client interface objects and your client application are consistent. Check your client application for parameter integrity.

00010211 Parameter Error: I4 > MAX. <direction> of IDL field [<parameter-name>] (<type-length>), location=<location>

Explanation An application parameter of IDL type I2 cannot be processed because its content is greater than the allowed maximum (+2147483647) for I2. See *IDL Data Types*.

The additional information <direction>, <parameter-name>, <type-length> and <location> is explained under error message 00010200.

Action Make sure the IDL file, generated client interface objects and your client application are consistent. Check your client application for parameter integrity.

00010212 Parameter Error: float < MIN. <direction> of IDL field [<parameter-name>] (<type-length>), location=<location>

Explanation An application parameter of IDL type F4 cannot be processed because its content is less than the allowed minimum (1.175494351e-38F) for F4. See *IDL Data Types*.

The additional information <direction>, <parameter-name>, <type-length> and <location> is explained under error message 00010200.

Action Make sure the IDL file, generated client interface objects and your client application are consistent. Check your client application for parameter integrity.

00010213 Parameter Error: float > MAX. <direction> of IDL field [<parameter-name>] (<type-length>), location=<location>

Explanation An application parameter of IDL type F4 cannot be processed because its content is greater than the allowed maximum (3.402823466e+38F) for F4. See *IDL Data Types*.

The additional information <direction>, <parameter-name>, <type-length> and <location> is explained under error message 00010200.

Action Make sure the IDL file, generated client interface objects and your client application are consistent. Check your client application for parameter integrity.

00010215 Parameter Error: float HUGE_VAL. <direction> of IDL field [<parameter-name>] (<type-length>), location=<location>

Explanation An application parameter of IDL type F4 cannot be processed because of conversation error HUGE_VAL. See C library `strtod()` for more information.

The additional information <direction>, <parameter-name>, <type-length> and <location> is explained under error message 00010200.

Action Make sure the IDL file, generated client interface objects and your client application are consistent. Check your client application for parameter integrity.

00010216 Parameter Error: double < MIN. <direction> of IDL field [<parameter-name>] (<type-length>), location=<location>

Explanation An application parameter of IDL type F8 cannot be processed because its content is less than the allowed minimum (2.2250738585072014e-308) for F8. See *IDL Data Types*.

The additional information <direction>, <parameter-name>, <type-length> and <location> is explained under error message 00010200.

Action Make sure the IDL file, generated client interface objects and your client application are consistent. Check your client application for parameter integrity.

00010217 Parameter Error: double > MAX. <direction> of IDL field [<parameter-name>] (<type-length>), location=<location>

Explanation An application parameter of IDL type F8 cannot be processed because its content is greater than the allowed maximum (1.7976931348623158e+308) for F8. See *IDL Data Types*.

The additional information <direction>, <parameter-name>, <type-length> and <location> is explained under error message 00010200.

Action Make sure the IDL file, generated client interface objects and your client application are consistent. Check your client application for parameter integrity.

**00010218 Parameter Error: double exponent over-, underflow. <direction> of IDL field
[<parameter-name>] (<type-length>), location=<location>**

Explanation An application parameter of IDL type F4 or F8 cannot be processed because of exponent over- or underflow. See *IDL Data Types*.

The additional information <direction>, <parameter-name>, <type-length> and <location> is explained under error message 00010200.

Action Make sure the IDL file, generated client interface objects and your client application are consistent. Check your client application for parameter integrity.

**00010219 Parameter Error: bad vdata provided. <direction> of IDL field
[<parameter-name>] (<type-length>), location=<location>**

Explanation An application parameter of IDL type AV, BV, KV or UV cannot be processed because incorrect data was applied, see *Using Variable-length Data Types AV, BV, KV and UV*.

The additional information <direction>, <parameter-name>, <type-length> and <location> is explained under error message 00010200.

Action Make sure the IDL file, generated client interface objects and your client application are consistent. Check your client application for parameter integrity.

**00010220 Parameter Error: bounds exceeds maximum. <direction> of IDL field
[<parameter-name>] (<type-length>), location=<location>**

Explanation An index of an application parameter for an IDL unbounded array (see *array-definition* under *Software AG IDL Grammar* in the IDL Editor documentation) cannot be processed because its value exceeds the maximum for upper bound.

The additional information <direction>, <parameter-name>, <type-length> and <location> is explained under error message 00010200.

Action Make sure the IDL file, generated client interface objects and your client application are consistent. Check your client application for parameter integrity.