Message Class 1001 - RPC Protocol

This class indicates an error occurred in the RPC Protocol. Messages of this class have the format:

1001*nnnn*

where 1001 is the message class, and

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

Overview of Messages

```
10010004 | 10010005 | 10010007 | 10010008 | 10010009 | 10010010 |
10010011 | 10010012 | 10010013 | 10010014 | 10010015 | 10010016 |
10010017 | 10010018 | 10010019 | 10010020 | 10010021 | 10010022 |
10010023 | 10010024 | 10010025 | 10010026 | 10010027 | 10010028 |
10010029 | 10010030 | 10010031 | 10010032 | 10010033 | 10010034 |
10010035 | 10010036 | 10010037 | 10010038 | 10010039 | 10010040 |
10010041 | 10010042 | 10010043 | 10010044 | 10010045 | 10010046 |
10010047 | 10010048 | 10010049 | 10010050 | 10010051 | 10010052 |
10010053 | 10010054 | 10010055 | 10010056 | 10010057 | 10010150 |
10010151 | 10010152 | 10010153 | 10010154 | 10010200 | 10010201 |
10010202 | 10010203 | 10010204 | 10010205 | 10010206 | 10010207 |
10010208 | 10010209 | 10010210 | 10010211 | 10010212 | 10010213 |
10010215 | 10010216 | 10010217 | 10010218 | 10010219 | 10010220 |
10012000 | 10012048 | 10012049 | 10012050
```

10010004 **Not enough memory to receive request**

Explanation The RPC server does not have enough memory to receive the client request.

Action If your server is a Natural RPC Server, increase the `MAXBUFF` memory parameter.

10010005 **Out of memory**

Explanation The RPC server is unable to allocate enough memory to fulfill the request.

Action Increase the memory resources of your RPC server.

10010007 **Internal error [Response <number> from RPC Server]**

Explanation The RPC server returned an internal error. A more specific error <*number*> may follow.

Action

1. See the trace on the server and/or client side and contact Software AG support.
2. See under error <*number*> for an explanation and action to take.

10010008 Unknown version

Explanation This error message is returned in the following situations:

1. The RPC protocol version you requested is not supported by your RPC server.
2. The server command is not known.
3. There is an error in the RPC protocol header.

Action Depending on the situation above, for

1. If possible, upgrade your RPC server or use only those data types supported by your RPC server in the IDL file.
2. See the trace on the server and/or client side and contact Software AG support.

10010009 Invalid format buffer

Explanation The format buffer within the RPC protocol could not be parsed.

Action Contact Software AG support.

10010010 Invalid value buffer

Explanation The value buffer within the RPC protocol could not be parsed.

Action Contact Software AG support.

10010011 Invalid packet type

Explanation The packet type shipped with the RPC protocol is not valid.

Action Contact Software AG support.

10010012 Authorization or initialization failed

Explanation One of the following occurred on the RPC server side:

1. A Natural RPC Server secured by Natural Security denied access because the Natural RPC Security ticket could not be evaluated correctly or is missing.
2. Initialization of a Natural RPC Server failed.
3. An EntireX z/OS CICS® RPC Server secured by CICS security denied the user's request.

Action Depending on the situation described above, check your security configuration on the server side.

10010013 No match with pattern

Explanation Natural RPC Server message.

Action Contact Software AG support.

10010014 Wrong Version of SYSRPC

Explanation Natural RPC Server message.

Action Contact Software AG support.

10010015 Recursive RPC tried

Explanation Natural RPC Server message.

Action Contact Software AG support.

10010016 Callee not found. [<Software AG IDL File Information>] [<RPC Server Message Appendix>]

Explanation The RPC server did not find the target customer server program. This message can be appended optionally by IDL information and an RPC Server Message Appendix, where

`<Software AG IDL File Information>` is either

- `Library=<Library>` see `library-definition` in the *Software AG IDL File* documentation
  or
- `Library=<Library>, Program=<Program>` see `program-definition` in the *Software AG IDL File* documentation

`<RPC Server Message Appendix>` depends on the target RPC server, for example:

- **CICS**
  For the CICS RPC Server (z/OS | z/VSE):
  - `RPC Server: CICS program <program> not found.`
- **COBOL**
  For the Batch RPC Server (z/OS | BS2000/OSD | z/VSE) if configured for COBOL(1), or *IMS RPC Server* if configured for COBOL(2), or *Micro Focus RPC Server*:
  - `RPC Server: COBOL program <program> not found.`
- **PL/I**
  For the Batch RPC Server (z/OS) if configured for PL/I(1) or *IMS RPC Server* if configured for PL/I(2):
  - `RPC Server: PL/I program <program> not found.`
- **Java**
  For the Java RPC Server (UNIX | Windows):
  - `RPC Server: java.lang.ClassNotFoundException: <class>`
  - `RPC Server: Method <method> not found in class <class> (java.lang.NoSuchMethodException: <class>.<method>())`
  - etc.
- **C**
  For the Batch RPC Server (BS2000/OSD) if configured for C(1), or EntireX RPC Server under UNIX | Windows:
  - `RPC Server: Library <library> not found.`
  - `RPC Server: Function (Entry Point) <function> not found.`
  - `RPC Server: Interface Object (Library <library> not found.`
  - `RPC Server: Interface Object (Entry Point) <function> not found.`
  - etc.

Action Depends on the target RPC server:

- **CICS**
  For the CICS RPC Server (z/OS | z/VSE): see *Locating and Calling the Target Server* in the z/OS | z/VSE RPC Server documentation on how to make the RPC server customer code available.
- **COBOL**
  For the Batch RPC Server (z/OS | BS2000/OSD | z/VSE) if configured for COBOL(1), or *IMS RPC Server* if configured for COBOL(2), or *Micro Focus RPC Server*: see *Locating and Calling the Target Server* in the z/OS | BS2000/OSD | z/VSE | Micro Focus RPC Server documentation on how to make the RPC server customer code available.
- **PL/I**
  For the Batch RPC Server (z/OS) if configured for PL/I(1), or *IMS RPC Server* if configured for PL/I(2): see *Locating and Calling the Target Server* in the Batch | IMS RPC Server documentation on how to make the RPC server customer code available.
- **Java**
  The action depends on the `<RPC Server Message Appendix>` returned. See *Generating a Java Server Interface Object* in the Java Wrapper documentation and *Using Package Names with the Java RPC Server* under UNIX | Windows in the Java RPC Server documentation for information.
- **C**
  For the Batch RPC Server (z/OS) if configured for C(1), or EntireX RPC Server under UNIX | Windows: the action depends on the `<RPC Server Message Appendix>` returned. See *Locating and Calling the Target Server* in the UNIX | Windows | BS2000/OSD RPC Server documentation on how to make the RPC server customer code available.
- etc.

Notes:

1. See `marshalling` parameter for the Batch RPC Server (z/OS | BS2000/OSD | z/VSE).
2. See `marshalling` parameter IMS RPC Server

10010017 Reserved for internal use

10010018 Abnormal termination during program execution. Lib=<Library>, Pgm=<Program>[,<Natural Error Info>]

Explanation The stub or the program/method called by the RPC server ended abnormally. This message can be appended by Natural error information:

```
[,(Natural Error Info: Program=<NAT Pgm>,Error=<NAT Error>,
                       Line=<NAT Line>,Status=<NAT Status>,Level=<NAT Level>)]
<NAT Pgm>       The Natural Program name which caused the error.
<NAT Error>     The Natural Error number.
<NAT Line>      The Line number within the erroneous program which caused the error.
<NAT Status>    See the Natural documentation.
<NAT Level>     See the Natural documentation.
```

Action Correct the program/method called by the RPC server.

10010019 Response from user program out of range

Explanation The program/method called by the RPC server can return application errors within the range 1 - 9999. The user error is outside of this range.

Action Check your program/method called by the RPC server and correct the error.

10010020 Reserved for internal use

10010021 Invalid header

Explanation The header within the RPC protocol is invalid.

Action See the trace on the server and/or client side and contact Software AG support.

10010022 Unknown server command

Explanation The RPC protocol contains an unknown server command.

Action Contact Software AG support.

10010023 Stub parameter definition error

Explanation The parameter definitions within the stub called by the RPC server are different. This error code is typical if clients send data using an older RPC protocol. The code will be detected by the RPC server to adjust its call information.

Action When working with the UNIX or Windows RPC server, no action is required if the error message appears just once per call. Multiple appearances indicate a general problem: re-generate your stubs with the template file server.tpl, compile and link them. If the error persists, contact Software AG support.

10010024 Parameter out of value range

Explanation The value range for a parameter is outside the defined range.
Example:
An I1 data type is mapped to a 2-byte integer type in a programming language environment where there is no 1-byte integer type. The range is defined in the interface specification, that is, the IDL file. See *Software AG IDL File*.
For I1 it is from -127 to 128. The mapping to a 2-byte integer permits a program to set values less than -127 or greater than 128 which is not allowed for I1.

Action Check program and correct error.

10010025 Parameter max len or array bound required

Explanation This error message is returned in the following situations:

1. The RPC server cannot handle variable-length data types such as AV, BV and KV without maximum lengths.
2. The RPC server cannot handle arrays with variable bounds without maximum upper bounds.

Action Depending on the situation above, for

1. Specify maximum values for the variable-length data types in the IDL file. See *Software AG IDL File*.
2. Specify maximum values for the upper bounds of the array in the IDL file.

10010026 Invalid string buffer

Explanation Internal error in the string buffer of the RPC protocol.

Action Contact Software AG support.

10010027 Parse error

Explanation The RPC server cannot parse the RPC protocol.

Action Contact Software AG support.

10010028 Library not found

Explanation The RPC server did not find the library or class given in the RPC protocol.

Action Check your configuration, server programs and stubs and the library name sent by the client.

10010029 Invalid eyecatcher

Explanation The header of the RPC protocol contains an invalid eyecatcher.

Action Contact Software AG support.

10010030 Parameter array bounds overflow

Explanation An overflow occurs with an unbounded array whose maximum array bounds are determined in the IDL file. See *Software AG IDL File*.

Action Check program and correct error.

10010031 String buffer item overflow of the RPC protocol

Explanation Internal error in string buffer.

Action Contact Software AG support.

10010032 Reserved for internal use

10010033 Invalid unbounded array

Explanation During marshalling/unmarshalling on the server side an error occurred when accessing the unbounded array.

Action One of the following may apply:

1. The unbounded array was destroyed by your server program. Check the program and correct the error.
2. An internal error occurred in the RPC server. Contact Software AG support.

10010034 Unbounded array does not match IDL file

Explanation The data type and attributes of the unbounded array sent to the server do not match the data type or attributes of the unbounded array within the server stub.

Action Re-generate, compile and link your server stub using the same IDL file as on the client side.

10010035 Conversation already open

Explanation The client application issues an open conversation RPC request having the conversation already opened.

Action Contact Software AG support.

10010036 Conversation already closed

Explanation The client application issues a close conversation, close conversation commit or conversational RPC request without having any open conversation.

Action Contact Software AG support.

10010037 Non-conv call within open conversation

Explanation The client application issues a non-conversational RPC request while having an open conversation.

Action Contact Software AG support.

10010038 Reserved for internal use

10010039 Error code converted to <encoding>, error code is <error code>

Explanation The RPC server does not reply in the correct encoding.

- <encoding> ASCII when the RPC server answers in ASCII but EBCDIC is expected

  EBCDIC when the RPC server answers in EBCDIC but ASCII is expected

- <error code> The error code replied by the Server.

Action Check that Internationalization is correctly configured for the broker. See *Internationalization with EntireX*.

10010040 Error code cannot be decoded

Explanation The RPC server does not reply in the correct encoding.

Action Check that Internationalization is correctly configured for the broker. See *Internationalization with EntireX*. Check that the server you are communicating with is an RPC server and not, for example, an ACI server.

10010041 Optional parameters not supported by RPC server

Explanation The RPC server does not support optional parameters. Optional parameters are supported by a Natural RPC Server. EntireX RPC servers do not support optional parameters.

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

10010042 **Maximum supported depth for Group levels is <level>**

Explanation The maximum level for groups is 99.

Action Correct your application.

10010043 **Change Password not supported by RPC server**

Explanation You tried to use the Change Password feature of the RPC protocol with an RPC server that does not support this feature.

Action Do not try to change the password with this RPC server.

10010044 **Natural RPC client stub required**

Explanation The IDL you are using contains features which require a Natural client interface object (stub).

Action Generate a Natural client interface object and use it with the RPC client.

10010045 **CICS ABEND <myabend> was issued**

Explanation This error indicates an application error code. Application error codes enable the RPC server to return customer-invented errors back to the RPC client in a standardized way without defining an error code field in the IDL.

Here the CICS RPC Server received a CICS `ABEND` code forced by the application server program using the `EXEC CICS ABEND ABCODE(`*`myabend`*`)`. The `ABEND` starts with a letter other than "A" and follows the rules for CICS user `ABEND` codes. The `ABEND` code (*`myabend`*) can be used in the RPC client to gain information about specific application error situations of the server.

For more information see *Aborting RPC Server Customer Code and Returning Error to RPC Client* (z/OS | z/VSE) in the CICS RPC Server documentation.

Action The error code can be used in RPC clients to test for specific application error codes.

10010046 **Inconsistent RPC request or Reliable RPC message**

Explanation The string buffer item "REL" shipped with the RPC protocol is not valid with RPC request.

Action Contact Software AG support.

10010047 Invalid value area

Explanation The value area in the server mapping file could not be parsed. A server mapping file is an EntireX Workbench file with extension .svm or .cvm. See *Server Mapping Files for COBOL*.

Action Contact Software AG support.

10010048 Inconsistent Reliable RPC message

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

Action Contact Software AG support.

10010049 Invalid packet type for Reliable RPC

Explanation The packet type shipped with the RPC protocol is not valid for the use of reliable RPC. The string buffer item "REL" is not valid for the packet type.

Action Contact Software AG support.

10010050 Illegal parameter direction

Explanation Reliable RPC allows only IDL direction `IN` parameters.

Action Update your IDL, i.e. remove any `OUT` and `INOUT` parameters, possibly re-generate your interface object and/or update your application, then retry the RPC call.

10010051 RPC request aborted by application program

Explanation The RPC client communicates with a CICS RPC Server. The RPC server programmer used `EXEC CICS ABEND CANCEL` together with `RPCUEX01` to abort the RPC server customer code. See *Using `EXEC CICS ABEND CANCEL`* under *Aborting RPC Server Customer Code and Returning Error to RPC Client* in the CICS RPC Server documentation.

Action This is an informational message. How the program continues depends on the application.

10010052 Server mapping not supported by RPC server [Library <library>, program <program>.]

Explanation The RPC server does not support client-side server mapping files (EntireX Workbench files with extension .cvm), see *Server Mapping Files for COBOL*). A wrong RPC server was used to execute the RPC request. For <library> see `library-definition` and for <program> see `program-definition` under *Software AG IDL Grammar*. The following RPC servers support client-side server mapping files:

- COBOL RPC servers: CICS RPC Server (z/OS, z/VSE) | Batch RPC Server (z/OS, z/VSE, BS2000/OSD) | *IMS RPC Server* | *Micro Focus RPC Server* | *CICS ECI RPC Server* | *IMS Connect RPC Server*.
- Natural RPC server.

Action Use an RPC server that supports client-side server mapping files.

10010053 Programming language <language> in server mapping not supported by RPC server. [Library <library>, program <program>.]

Explanation The RPC server does not support the programming language (Natural, COBOL etc.) in the server mapping file (EntireX Workbench files with extension .svm or .cvm). For <library> see `library-definition` and for <program> see `program-definition` under *Software AG IDL Grammar*. One of the following situations occurred:

1. At design time with the EntireX Workbench, a server mapping file containing COBOL mapping was created with COBOL plug-in *IDL Extractor for COBOL* or *COBOL Wrapper* (Generate RPC Server). See also *Server Mapping Files for COBOL*.
   At runtime, however, the RPC server used was not a COBOL RPC server.
2. At design time with the EntireX Workbench, a server mapping file containing Natural mapping was created with Natural plug-in *IDL Extractor for Natural* or *Natural Wrapper* (Generate RPC Server). See also *Server Mapping Files for Natural*.
   At runtime, however, the RPC server used was not a Natural RPC server.

Action

- For situation 1, use one of the following COBOL RPC servers: CICS RPC Server (z/OS, z/VSE) | Batch RPC Server (z/OS, z/VSE, BS2000/OSD) | *IMS RPC Server* | *Micro Focus RPC Server* | *CICS ECI RPC Server* | *IMS Connect RPC Server*.
- For situation 2, use a Natural RPC server.

10010054 Interface type <interface-type> in server mapping not supported by RPC server. [Library <library>, program <program>.]

Explanation The interface type in the server mapping (Batch, DFHCOMMAREA, IMPS MPP (IMS Connect) etc.) is not supported by the RPC server and/or the platform the RPC server is running on. For <library> see `library-definition` and for <program> see `program-definition` under *Software AG IDL Grammar*.

At design time with the EntireX Workbench, a server mapping file (Workbench file with extension .svm or .cvm) for COBOL with a specific interface type was created with a COBOL plug-in: *IDL Extractor for COBOL* or *COBOL Wrapper* (Generate RPC Server).

To call a server successfully, the RPC server used must support the interface type of the COBOL server. The table below gives an overview of possible combinations of an interface type and a supporting RPC server:

Interface Type	Supported by EntireX Adapter	Supported by RPC Server								
		z/OS			UNIX/Windows			BS2000/OSD	z/VSE	
		CICS	Batch	IMS	CICS ECI	Micro Focus	IMS Connect	Batch	CICS	Batch
CICS with `DFHCOMMAREA` Calling Convention (Extractor | Wrapper)	x	x			x				x	
CICS with `DFHCOMMAREA` Large Buffer Interface (Extractor | Wrapper)		x							x	
CICS with Channel Container Calling Convention (Extractor | Wrapper)		x								
Batch with Standard Linkage Calling Convention (Extractor | Wrapper)			x	x				x		x
Micro Focus with Standard Linkage Calling Convention (Extractor | Wrapper)						x				
IMS BMP with Standard Linkage Calling Convention (Extractor | Wrapper)				x						
IMS MPP Message Interface (IMS Connect) (Extractor)	x						x			

Action Probably a user error. For example, a CICS interface type was used with a Batch RPC Server.

10010055 **Sent IDL type N/NU/P/PU too long, RPC server supports <number-of-digits> digits max**

Explanation The length of a numeric IDL type, one of N/NU/P/PU sent by the RPC client is too long for the target programming language and environment where the RPC server is running.

Action Update your IDL. Re-generate your interface object and/or update your application, then retry the RPC call.

10010056 **Configuration issue: For program <program> library <library> the RPC server has a server-side mapping and received a conflicting client-side mapping from the RPC client**

Explanation The RPC client sent a client-side mapping file (EntireX Workbench file with extension .cvm), but a server-side mapping file (.svm) also still exists in the server mapping container for <library> see `library-definition` and for <program> see `program-definition` under *Software AG IDL Grammar*. See *Server-side Mapping Files in the RPC Server* in the RPC server documentation for z/OS (CICS, Batch, IMS) | Micro Focus | CICS ECI | IMS Connect | BS2000/OSD | z/VSE (CICS, Batch). Usage of server mapping files is mutually exclusive: use either client-side or server-side mapping files. See *Server Mapping Files for COBOL*.

Action If you migrate server-side mapping files to client side mapping files, you may have forgotten *Step 2: Remove the Server-side Mapping Files on Target RPC Server*. See *Migrating Server Mapping Files* under *Server Mapping Files for COBOL* in the EntireX Workbench documentation.

10010057 **Version mismatch: For program <program> (library <library>) the RPC server found the unsupported version <svm-version> in the server mapping**

Explanation The server mapping for the IDL library and IDL program contained a server mapping version not supported by the RPC server. A server mapping file is an EntireX Workbench file with extension .svm or .cvm. See *Server Mapping Files for COBOL*.

Action Contact Software AG support for RPC server upgrade.

10010150 **Format Buffer Check: direction does not match. Request of IDL field (<type-length>), location=<location>**

Explanation The EntireX RPC server detected an inconsistency between the deployed server-side mapping (see *Server-side Mapping* under *Server Mapping Files for COBOL* in the EntireX Workbench documentation) and the IDL file with which the RPC client was built. The `direction-attribute` (`IN`, `OUT`, `INOUT`) of a parameter (see `attribute-list` under *Software AG IDL Grammar*) defined in the IDL file is different to the `direction-attribute` in the related server-side mapping.

Possible causes:

- You are calling an existing COBOL server and have reextracted the IDL and server-side mapping without redeploying the server-side mapping file. See Scenario *Calling an Existing COBOL Server* under z/OS (CICS, Batch, IMS) | Micro Focus | BS2000/OSD | z/VSE (CICS, Batch), and *Server Mapping Deployment Wizard* in the EntireX Workbench documentation.
- You are writing a new COBOL server and have rebuilt the RPC server with a modified IDL file using the *COBOL Wrapper*, which updates the server-side mapping file, but the updated server-side mapping file is not redeployed. See Scenario *Writing a New COBOL Server* under z/OS (CICS, Batch, IMS) | Micro Focus RPC Server | BS2000/OSD | z/VSE (CICS, Batch), and and *Server Mapping Deployment Wizard* in the EntireX Workbench documentation.
- etc.

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

- `<type-length>` matches the IDL data type; see *IDL Data Types*
- `<location>` contains the hierarchy of IDL nodes, separated by a colon

Example: `IDL field (NU14.0), location=3.2.1`.

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`'; 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 and the deployed server-side mapping file are consistent. Possibly redeploy the server-side mapping file, see *Deploying Server-side Mapping Files to the RPC Server* in the RPC server documentation for z/OS (CICS, Batch, IMS) | Micro Focus | CICS ECI | IMS Connect | BS2000/OSD | z/VSE (CICS | Batch). If the problem persists, reextract or rewrap and then redeploy.

10010151 Format Buffer Check: data type does not match. Request of IDL field (<type-length>), location=<location>

Explanation The EntireX RPC server detected an inconsistency between the deployed server-side mapping (see *Server-side Mapping* under *Server Mapping Files for COBOL* in the EntireX Workbench documentation) and the IDL file with which the RPC client was built. The sequence of IDL data types (see *IDL Data Types* under *Software AG IDL File* in the IDL Editor documentation) defined in the IDL file is different to the sequence in the related server-side mapping. See error message 10010150 for possible causes of this situation and additional information for `<type-length>` and `<location>`.

Action Make sure the IDL file and the deployed server-side mapping file are consistent. Possibly redeploy the server-side mapping file, see *Deploying Server-side Mapping Files to the RPC Server* in the RPC server documentation for z/OS (CICS, Batch, IMS) | Micro Focus | CICS ECI | IMS Connect | BS2000/OSD | z/VSE (CICS | Batch). If the problem persists, reextract or rewrap and then redeploy.

10010152 Format Buffer Check: length is not equal. Request of IDL field (<type-length>), location=<location>

Explanation The EntireX RPC server detected an inconsistency between the deployed server-side mapping (see *Server-side Mapping* under *Server Mapping Files for COBOL* in the EntireX Workbench documentation) and the IDL file with which the RPC client was built. The length definition of a parameter (see `simple-parameter-definition` under *Software AG IDL Grammar*) defined in the IDL file is different to the length definition in the related server-side mapping. See error message 10010150 for possible causes of this situation and additional information for `<type-length>` and `<location>`.

Action Make sure the IDL file and the deployed server-side mapping file are consistent. Possibly redeploy the server-side mapping file, see *Deploying Server-side Mapping Files to the RPC Server* in the RPC server documentation for z/OS (CICS, Batch, IMS) | Micro Focus | CICS ECI | IMS Connect | BS2000/OSD | z/VSE (CICS | Batch). If the problem persists, reextract or rewrap and then redeploy.

10010153 Format Buffer Check: attributes not matching. Request of IDL field (<type-length>), location=<location>

Explanation The EntireX RPC server detected an inconsistency between the deployed server-side mapping (see *Server-side Mapping* under *Server Mapping Files for COBOL* in the EntireX Workbench documentation) and the IDL file with which the RPC client was built. The attributes of a parameter (see `attribute-list` under *Software AG IDL Grammar*) defined in the IDL file are different to the attributes in the related server-side mapping. See error message 10010150 for possible causes of this situation and additional information for `<type-length>` and `<location>`.

Action Make sure the IDL file and the deployed server-side mapping file are consistent. Possibly redeploy the server-side mapping file, see *Deploying Server-side Mapping Files to the RPC Server* in the RPC server documentation for z/OS (CICS, Batch, IMS) | Micro Focus | CICS ECI | IMS Connect | BS2000/OSD | z/VSE (CICS | Batch). If the problem persists, reextract or rewrap and then redeploy.

10010154 Format Buffer Check: indices not matching. Request of IDL field (<type-length>), location=<location>

Explanation The EntireX RPC server detected an inconsistency between the deployed server-side mapping (see *Server-side Mapping* under *Server Mapping Files for COBOL* in the EntireX Workbench documentation) and the IDL file with which the RPC client was built. Either the number of dimensions (one-, two- three-dimensional) or the upper bounds (see `array-definition`) do not match. See error message 10010150 for possible causes of this situation and additional information for `<type-length>` and `<location>`.

Action Make sure the IDL file and the deployed server-side mapping file are consistent. Possibly redeploy the server-side mapping file, see *Deploying Server-side Mapping Files to the RPC Server* in the RPC server documentation for z/OS (CICS, Batch, IMS) | Micro Focus | CICS ECI | IMS Connect | BS2000/OSD | z/VSE (CICS | Batch). If the problem persists, reextract or rewrap and then redeploy.

10010200 RPC 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 and, if it exists, the server-side mapping file and your server application are consistent. See *Server Mapping Files for COBOL* in the EntireX Workbench documentation. Check your application for parameter integrity.

10010201 **RPC 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 10010200.

Action Make sure the IDL file and, if it exists, the server-side mapping file and your server application are consistent. See *Server Mapping Files for COBOL* in the EntireX Workbench documentation. Check your application for parameter integrity.

10010202 **RPC 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 10010200.

Action Make sure the IDL file and, if it exists, the server-side mapping file and your server application are consistent. See *Server Mapping Files for COBOL* in the EntireX Workbench documentation. Check your application for parameter integrity.

10010203 **RPC 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 10010200.

Action Make sure the IDL file and, if it exists, the server-side mapping file and your server application are consistent. See *Server Mapping Files for COBOL* in the EntireX Workbench documentation. Check your application for parameter integrity.

10010204 **RPC 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 10010200.

Action Make sure the IDL file and, if it exists, the server-side mapping file and your server application are consistent. See *Server Mapping Files for COBOL* in the EntireX Workbench documentation. Check your application for parameter integrity.

10010205 **RPC 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 10010200.

Action Make sure the IDL file and, if it exists, the server-side mapping file and your server application are consistent. See *Server Mapping Files for COBOL* in the EntireX Workbench documentation. Check your application for parameter integrity.

10010206 **RPC 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 10010200.

Action Make sure the IDL file and, if it exists, the server-side mapping file and your server application are consistent. See *Server Mapping Files for COBOL* in the EntireX Workbench documentation. Check your application for parameter integrity.

10010207 RPC 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 10010200.

Action Make sure the IDL file and, if it exists, the server-side mapping file and your server application are consistent. See *Server Mapping Files for COBOL* in the EntireX Workbench documentation. Check your application for parameter integrity.

10010208 RPC 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 10010200.

Action Make sure the IDL file and, if it exists, the server-side mapping file and your server application are consistent. See *Server Mapping Files for COBOL* in the EntireX Workbench documentation. Check your application for parameter integrity.

10010209 RPC 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 10010200.

Action Make sure the IDL file and, if it exists, the server-side mapping file and your server application are consistent. See *Server Mapping Files for COBOL* in the EntireX Workbench documentation. Check your application for parameter integrity.

10010210 RPC 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 10010200.

Action Make sure the IDL file and, if it exists, the server-side mapping file and your server application are consistent. See *Server Mapping Files for COBOL* in the EntireX Workbench documentation. Check your application for parameter integrity.

10010211 RPC 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 10010200.

Action Make sure the IDL file and, if it exists, the server-side mapping file and your server application are consistent. See *Server Mapping Files for COBOL* in the EntireX Workbench documentation. Check your application for parameter integrity.

10010212 RPC 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 10010200.

Action Make sure the IDL file and, if it exists, the server-side mapping file and your server application are consistent. See *Server Mapping Files for COBOL* in the EntireX Workbench documentation. Check your application for parameter integrity.

10010213 RPC 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 10010200.

Action Make sure the IDL file and, if it exists, the server-side mapping file and your server application are consistent. See *Server Mapping Files for COBOL* in the EntireX Workbench documentation. Check your application for parameter integrity.

10010215 RPC 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 10010200.

Action Make sure the IDL file and, if it exists, the server-side mapping file and your server application are consistent. See *Server Mapping Files for COBOL* in the EntireX Workbench documentation. Check your application for parameter integrity.

10010216 RPC 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 10010200.

Action Make sure the IDL file and, if it exists, the server-side mapping file and your server application are consistent. See *Server Mapping Files for COBOL* in the EntireX Workbench documentation. Check your application for parameter integrity.

10010217 RPC 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 10010200.

Action Make sure the IDL file and, if it exists, the server-side mapping file and your server application are consistent. See *Server Mapping Files for COBOL* in the EntireX Workbench documentation. Check your application for parameter integrity.

10010218 RPC 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 10010200.

Action Make sure the IDL file and, if it exists, the server-side mapping file and your server application are consistent. See *Server Mapping Files for COBOL* in the EntireX Workbench documentation. Check your application for parameter integrity.

10010219 RPC 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 10010200.

Action Make sure the IDL file and, if it exists, the server-side mapping file and your server application are consistent. See *Server Mapping Files for COBOL* in the EntireX Workbench documentation. Check your application for parameter integrity.

10010220 **RPC 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 10010200.

Action Make sure the IDL file and, if it exists, the server-side mapping file and your server application are consistent. See *Server Mapping Files for COBOL* in the EntireX Workbench documentation. Check your application for parameter integrity.

10012000 **Reserved for internal use**

10012048 **Reserved for internal use**

10012049 **Reserved for internal use**

10012050 **Reserved for internal use**