

FAQ

- We normally force our applications to use `InvariantCulture`, it seems FA .NET spawns each thread with machine culture. Is there any setting to set `CultureInfo` to a default value instead of machine culture?
- Is there any option to switch on/off validation of tag value type (e.g. if a received message contains `38=ten`)?
- Is it possible to start more than one instance of FIX Antenna C++?
- Is it possible to create more than one instance of FIX Antenna C++ using the same logging directory?
- How to create FIX50, FIX50SP1 or FIX50SP2 session?
- May I put session log files to the log directory and expect that FIX Antenna C++ will analyze them when a new session with the same parameters is created?
- FIX Antenna C++ was incorrectly terminated. I don't want the session to be restored on next start. What should I do for a clean start?
- How can I do the customization (add user-defined fields, messages or change field, and message definitions)?
- How to fix the "Repeating group fields out of order" error?
- How to remove Repeating Group?
- How to change the order of tags in a FIX message?
- Is it possible to send messages in `onLogonEvent` handler?
- How to extend FIX message with a repeating group defined in FIX protocol?
- Is it possible to use `FixMessage` and `FixGroup` after `FixEngine::destroy()` call?
- What is the format of FIX dictionary (`fixdict.xml`) file?
- Is there a way to enable 'automatic reconnection' in the case that the FIX-sessions drops? I want to reconnect as soon as the server or the connection is available again.
- How can I get a message (`std::vector<const Engine::TagValue*>`) as a string or set of fields? I have found only `getFields()` function, but it isn't implemented. I am also interested in how can I get any field by tag. For example, I am trying to call `getField(279)` function and then `toString()` and I have got error message "Not supported: CharFiledValue::toString Tag 279." How should I convert or get tag values?
- How to create multiple sessions with the same Sender/Target?
- What is Session Qualifier?
- How to configure validation for non-user tags?
- Is there any end of day command, which resets sequence numbers, truncates database etc. for specific session on FA.NET?
- How to fix a problem when an application cannot receive multicast data (IGMP reports) through a VPN connection
- I am getting the 'Exception from HRESULT: 0x80131515' when I try to run my application built with `FixAntenna .Net` on Windows 7 (and higher).
- May I use environment variables as values of configuration parameters
- Are there any supported application accelerators other than Myricom DBL
- FIX Engine doesn't start with `getaddrinfo()` failed error.
- Method `getGroup()` throw the exception "Tag is not defined" for a message with FIX50+ version

We normally force our applications to use `InvariantCulture`, it seems FA .NET spawns each thread with machine culture. Is there any setting to set `CultureInfo` to a default value instead of machine culture?

It's not Fix Antenna issue. It's a common behavior of .Net threading api (https://docs.microsoft.com/en-us/dotnet/api/system.globalization.cultureinfo.currentculture?redirectedfrom=MSDN&view=net-5.0#System_Globalization_CultureInfo_CurrentCulture). So if you are using .Net Framework >= 4.5, you can use `DefaultThreadCurrentCulture` property, otherwise, you can manually firstly check (to avoid locks) and set the locale in each FA callback.

Is there any option to switch on/off validation of tag value type (e.g. if a received message contains `38=ten`)?

Yes, there is the option

```
Session.Default.Validation.VerifyTagsValues
```

in `engine.properties`.

You also can use

```
SessionExtraParameters::ValidationParameters::verifyTagsValues_
```

from FIX Antenna's C++ API.

This option enables validation of field values format.

It works only if validation enabled (`SessionExtraParameters::ValidationParameters::isEnabled_` is switched on).

Is it possible to start more than one instance of FIX Antenna C++?

Yes, it is. Inside **one process** it is possible to start only **one** instance of the FIX Engine. The number of processes with FIX Engine started is **not limited**.

Is it possible to create more than one instance of FIX Antenna C++ using the same logging directory?

Yes. However, it is not recommended. Please note `LogFileName` parameter should be unique.

How to create FIX50, FIX50SP1 or FIX50SP2 session?

The following code demonstrates how to create a session for FIX50, FIX50SP1 or FIX50SP2 protocol.

```
/* Creating FIX50 session. Replace FIX50 with FIX50SP1 or FIX50SP2 if required. */
Engine::Session* session = Engine::FixEngine::singleton()->createSession(&appl, "sender", "target", Engine::FIX50
);
```

May I put session log files to the log directory and expect that FIX Antenna C++ will analyze them when a new session with the same parameters is created?

Yes. FIX Antenna C++ loads all session information on session creation time.

FIX Antenna C++ was incorrectly terminated. I don't want the session to be restored on next start. What should I do for a clean start?

Delete session log files from logs folder by mask:

```
<sender>-<target>*.*
```

How can I do the customization (add user-defined fields, messages or change field, and message definitions)?

User-Defined Fields (the tag numbers 5000 to 9999) are handled like ordinary fields. User-defined FIX messages (e.g. U1, U2) are also handled like ordinary messages. All other messages can be customized in the way explained in the [Protocol customization quick start](#)

How to fix the "Repeating group fields out of order" error?

Example:

```
8=FIX.4.3 | 9=285 | 35=X | 49=FIXERBRIDGETRD1 | 34=1 | 57=B2BTEST1 | 52=99990909-17:17:17 | 268=4 | 269=2 | 270=9539.500000 | 271=1 |
272=20041208 | 273=13:34:10.000 | 290=1 | 269=4 | 27 0 =9539.500000 | 271=4294967295 | 272=20041208 | 273=13:34:10.000 | 290=1 | 269=Z |
270=9528.000000 | 271=401 | 290=1 | 269=Y | 270=0.000000 | 271=0 | 290=1 | 10=019 |
```

```
[ERROR] Repeating group fields out of order [RefSeqNum: 1, RefTagID: 269, RefMsgType: X]
```

Solution: The reason of this problem cannot be detected easily. The required fields of both FIX message itself and (AdditionalFields) group must be filled out with corresponding values and any gap or change in tag order leads to an error (it is true for both standard and custom messages). Therefore, both standard and additional group fields must be checked for validity and then processed again. Extra attention must be paid to the order of fields in the repeating FIX group. For this particular example: tag 290 is placed before tag 269 (wrong tag order). Place the 290 tag and its value after the tag 269 and process the message again to solve the problem.

How to remove Repeating Group?

Use one of the following methods. "tag" parameter is tag of leading field (field which contains the size of the group).

- `FIXMessage::remove(int tag)` to remove group from FIX message
- `TagValue::remove(int tag)` to remove group from repeating group entry or from FIX message
- `FIXGroup::remove(int tag)` to remove group from repeating group entry

How to change the order of tags in a FIX message?

The order of tags in a FIX message is defined by their position in FIX protocol [dictionaries](#). For example, if you would like to change the default order of tags 44 ([Price](#)) and 15 ([Currency](#)) in [Advertisement messages](#) (35=7), then swap the position of tags under corresponding section:

```
<msgdef msgtype="7" name="Advertisement">
...
  <field tag="44" name="Price"/>
  <field tag="15" name="Currency"/>
...
</msgdef>
```

to:

```
<msgdef msgtype="7" name="Advertisement">
...
    <field tag="15" name="Currency"/>
    <field tag="44" name="Price"/>
...
</msgdef>
```

Is it possible to send messages in onLogonEvent handler?

When session-acceptor receives a Logon message, the onLogonEvent is fired. Sending messages in this event handler causes session none graceful termination. The reason is that the FIX protocol requires the first message to be Logon. It is prohibited to send messages in onLogonEvent handler.

How to extend FIX message with a repeating group defined in FIX protocol?

The AdditionalFields mechanism requires repeating the group specification fully. It is not enough to add a repeating group leading tag to add this group to the FIX message - all tags of this group have to be enumerated as well. For example the correct extension of the BF message is: Validation. AdditionalFields = FIX44:BF:386(336,625?)

Is it possible to use FixMessage and FixGroup after FixEngine::destroy() call?

It is prohibited to continue using FIXMessage and FixGroup objects after engine::destroy() call, you may only destroy them.

What is the format of FIX dictionary (fixdict.xml) file?

Format of the FIX dictionary file (a.k.a fixdict.xml) is defined by set of XSD files:

- fixdic.xsd
- fixdic_abstract.xsd
- fixdics_operations.xsd

Format of the conditionally required expression is similar to Pascal syntax. Example:

```
T$1414 > 0 and existstags(T$1415) or T$1415 in ('A', 'B')
```

Set of available functions/operators:

- existstags(T\$<tag>, T\$<tag>, ...)
- in
- =, !=, <, >, >=, <=
- and, or, not
- T\$<tag> - returns field value. Example: T\$123.

Sample of XML protocol file: [Creating protocol customization](#)

Is there a way to enable 'automatic reconnection' in the case that the FIX-sessions drops? I want to reconnect as soon as the server or the connection is available again.

You can set reconnect max tries `reconnectMaxTries_` session parameter to -1. This will force FIX Antenna to reconnect infinitely.

To interrupt reconnection call `Session::disconnectNonGracefully` method.

Reconnection also can be configured within engine.properties file: [Settings of Session Reconnection](#)

How can I get a message (std::vector<const Engine::TagValue*>) as a string or set of fields? I have found only getFields() function, but it isn't implemented. I am also interested in how can I get any field by tag. For example, I am trying to call getField(279) function and then toString() and I have got error message "Not supported: CharFiledValue::toString Tag 279." How should I convert or get tag values?

There is currently no method to convert TagValue to string. Only FIXMessage can be represented as string.

When you are receiving "Not supported: CharFiledValue::toString" you should change your code to call `FIXMessage::getAsChar`. In general, you should check the type of the field in the FIX dictionary file and use the appropriate getter method. Almost all FIX types and C++ types have same names. Exception is `FIX::float`. In C++ it is `Engine::Decimal`.

How to create multiple sessions with the same Sender/Target?

In version 2.10.15.5 of FIX Antenna and newer, you should first create Engine::SessionId structure by passing Sender, Target and unique Session Qualifier to its constructor. Then call one of the overloaded methods createSession() which accepts Engine::SessionId structure as one of the arguments. See section "Session identifier" of the documentation for more details

What is Session Qualifier?

Session Qualifier is an extra identifier, which together with SenderCompID and TargetCompID uniquely identify session. In particular, this allows the user creating several sessions with the same Sender/Target pair. See the section "Session identifier" of the documentation for more details

How to configure validation for non-user tags?

FIX Engine has 3 modes for processing undefined and user-defined tags. Tags from 5000 are considered as user-defined tags which have specific validation rules.

1. Truncate all undefined tags (the best performance):

engine.properties

```
IgnoreUnknownFields = true
ProhibitUnknownTags = true
```

The **'true'** for the **IgnoreUnknownFields** parameter means that unknown tags will be removed by FIX Engine. This increases performance. The **'true'** for the **ProhibitUnknownTags** parameter means that the FIX engine will reject messages with unknown tags if MessageMustBeValidated = true



To avoid truncation, the tags should be defined in the FIX dictionary for the corresponding message. The session incoming logs (*.in) contain received messages without any truncation (as it was received from socket).

2. Reject undefined non-user (i.e. less than 5000) tags:

engine.properties

```
IgnoreUnknownFields = false
ProhibitUnknownTags = true
```

The **'false'** for the **IgnoreUnknownFields** parameter means that the unknown user-defined tags will remain in the message consuming memory and reducing performance.

The **'true'** for the **ProhibitUnknownTags** parameter means that the FIX engine will reject messages with unknown tags if MessageMustBeValidated = true

3. Do not validate and do not reject undefined and user-defined tags:

engine.properties

```
IgnoreUnknownFields = false
ProhibitUnknownTags = false
```

Is there any end of day command, which resets sequence numbers, truncates database etc. for specific session on FA.NET?

Yes, there is Session::ResetSeqNum method. If you wish to reset sequences, you should call:

```
session.ResetSeqNum( SeqNumResetStrategy.ResetSeqNum )
```

This method creates new message storage file and resets message sequence numbers to initial value (in=1, out=1).

How to fix a problem when an application cannot receive multicast data (IGMP reports) through a VPN connection

This problem can be occurred on a client computer that is running Windows 7, Windows Server 2008 R2, Windows Vista, or Windows Server 2008. To resolve this issue you can download a hotfix from Microsoft, for more information read this articles:

<https://support.microsoft.com/en-us/topic/igmp-v1-membership-report-packets-are-sent-to-incorrect-destinations-on-a-computer-that-is-running-windows-server-2008-windows-vista-windows-7-or-windows-server-2008-r2-4f07dcab-dbba-33cd-872e-b838e0d7edda>

<https://social.technet.microsoft.com/Forums/windows/en-US/bab679df-5607-4399-ac6c-197f379fb9a9/windows-7-no-multicast-igmp-reports-on-vpn-connection?forum=w7itpronetworking>

I am getting the 'Exception from HRESULT: 0x80131515' when I try to run my application built with FixAntenna .Net on Windows 7 (and higher).

You need to unlock the FixAntenna library.

To do it, please, refer to the article:

[https://docs.microsoft.com/en-us/previous-versions/ee890038\(v=vs.110\)?redirectedfrom=MSDN](https://docs.microsoft.com/en-us/previous-versions/ee890038(v=vs.110)?redirectedfrom=MSDN)

May I use environment variables as values of configuration parameters

Yes, please see [How to configure properties with environment variables](#) article for more details.

Are there any supported application accelerators other than Myricom DBL

In order to use [Myricom DBL](#) it is required to utilize their specific API. Because of that, we [added and described its distinct support](#).

[Solarflare](#) cards do not use separate API and their drivers work transparently for the application. Thus [Solarflare OpenOnload](#) does not require separate code support and can be used with our products.

FIX Engine doesn't start with getaddrinfo() failed error.

If there is an error message contains:

```
getaddrinfo(<hostName>) failed: <error code>
```

Most likely this error means a DNS configuration issue, so it should be checked.

Also, it can be fixed by resolving the hostname in **hosts** configuration file.

Method getGroup() throw the exception "Tag is not defined" for a message with FIX50+ version

For example, one wants to get access to the group for a message with version FIX 5.0 or higher (i.e. for a version using transport dictionaries). In the example below, it's a group [NoMDEntries \(268\)](#) for message [Market Data - Incremental Refresh \(X\)](#)

```
Engine::Session* session= Engine::FixEngine::singleton()->createSession(&application, "TargetCompID2",
"SenderCompID2",
    Engine::FIX50,
    &params, Engine::persistent_storageType, Engine::FIXT11_TCP);
std::auto_ptr<Engine::FIXMessage> pMessage2(session->newSkel("X"));
try
{
    Engine::FIXGroup* pMessage2->getGroup(FIXFields::NoMDEntries);
}
catch (const Utils::Exception& ex)
{
    // ...
}
```

This code will throw an exception: "Tag is not defined: 268"

Root cause:

By default, if the application version isn't passed to [Engine::Session::newSkel \(...\)](#) method then FIX Antenna uses a transport dictionary (for example **fixdict 11.xml**) for message creation. However, in the transport dictionary, there are no business-level tags such as [NoMDEntries \(268\)](#) so an exception about missing tag is thrown.

Solution:

The fix protocol version of application-level tags should be passed for creating or parsing messages for FIX.5.0 versions or higher.

The user should pass a specific [Engine::FIXVersion](#) or result of [Engine::Session::getAppVer\(\)](#) as an argument. For example:

```
std::auto_ptr<Engine::FIXMessage> pMessage2(pSI2->newSkel("X"), session->getAppVer());
```