

# FIX Antenna C++/.NET dictionaries format

- [Overview](#)
- [Dictionary structure](#)
  - [Standalone dictionary structure](#)
  - [Additional dictionary specifics](#)
- [How to manage list of dictionaries in FIXAntenna based applications](#)
- [Dictionary customization](#)
- [How to check that dictionary doesn't have errors](#)

## Overview

FIX dictionary is an XML-file which describes [FIX messages](#), fields and groups. FIX dictionary can be:

- self-sufficient dictionary which describes the [FIX protocol](#);



FIX Antenna JAVA dictionaries have the same format as FIX Antenna C++/.NET dictionaries

- additional dictionary which describes changes to be applied to another existing dictionary.



FIX Antenna JAVA does not support additional dictionaries.

One or more [FIX protocols](#) are required for [FIX session](#) work.

FIX dictionaries are combined into parsers. Parsers are used to validate incoming messages, i.e. they check if messages satisfy defined protocols.

When FIX Antenna is started, all dictionaries from [DictionariesFilesList](#) are loaded. If dictionaries are standard, predefined standard parsers (like FIX40; FIX41; FIX42; FIX43; FIX44; FIXT11) are created basing on them. If any dictionary from [DictionariesFilesList](#) is custom, parser is not automatically created and therefore such parsers should be listed manually in [AdditionalParsersList](#) parameter.

Overall there should be one session protocol (protocol which describes [messages of session level](#)) and one or more application protocols (protocols which describe [messages of application level](#)) defined in common parsers list (i.e. in standard parsers list and [additional parsers list](#) together).

Description for all standard FIX protocols can be found on [fixopaedia](#).

## Dictionary structure

### Standalone dictionary structure

Below is the common structure of the standalone dictionary with used tags and attributes description.

Structure	Description
<fixdic>	Root tag of the dictionary.
@xmlns	xmlns="http://www.b2bits.com/FIXProtocol"
@xmlns:xsi	xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
@xsi:schemaLocation	xsi:schemaLocation="http://www.b2bits.com/FIXProtocol fixdic.xsd"
@id	ID of the used dictionary. Valid values are: <ul style="list-style-type: none"><li>• FIX40</li><li>• FIX41</li><li>• FIX42</li><li>• FIX43</li><li>• FIX44</li><li>• FIX50</li><li>• FIX50SP1</li><li>• FIX50SP2</li><li>• FIX50SP2EP</li><li>• FIXT11</li><li>• ID of the custom dictionary (refer to <a href="#">How to use custom dictionaries with FIXEdge</a> for details)</li></ul>

@fixversion	FIX version of the used dictionary. E.g.: 4.4, 4.2, T1.1, etc.
@title	The name of the dictionary. E.g.: FIX 5.0 SP2 EP, <a href="#">FIX 4.4 (with errata 20030618)</a>
@version	Dictionary version.
@date	Date of the last dictionary update.
<typelist>	Section which describes types used in the dictionary.
<typedef>	Definition of the type.
@type	Type name. E.g.: Percentage, Amt, char
@extends	Specified if the type extends one of existing types. E.g.: <ul style="list-style-type: none"> <li>• &lt;typedef type="Percentage" extends="float"&gt;</li> <li>• &lt;typedef type="Boolean" extends="char"&gt;</li> </ul>
@valuetype	Type of the value (specifies either type is multiple value type and type of the underlying value).
@displayname	Name to be displayed (specifies either type is multiple value type and type of the underlying value).
<descr>...</descr>	Description of the type.
</typedef>	-
</typelist>	-
<fielddic>	Section which describes fields used in the dictionary.
<valblockdef>	Definition of the block of element's values. Can be zero or more elements. E.g. for countries, currencies, products, securities, etc.
@id	Unique identifier of the block.
@name	Name of the block.
<item>	Item of the block - name of the item.
@val	Serial number of the item.
@id	Unique identifier of the item. Id is an optional attribute and is used when description of an item value is longer than 4 words or has any special characters.
<msgref>	Reference to the message where mentioned item is used - name of the message.
@msgtype	Type of the reference message.
</msgref>	-
</item>	-
<range>	Range of values to be supported - purpose of the range.
@minval	Minimum value to be supported.
@maxval	Maximum value to be supported.
@type	Type of values.
</range>	-
<multi>	List of items. The field can contain multiple items separated by space.
<item>...</item>	The structure is the same as described for <i>fixdic/fielddic/valblockdef/item</i> .
</multi>	-
<descr>...</descr>	Description.
</valblockdef>	-
<fielddef>	Definition of the field.
@tag	Tag of the field.
@name	Name of the field.
@type	Type of the field.
@transport	Transport.

	@lenfield	Only for fielddef with type="data" or type="XMLData"
	<alias>...</alias>	Alternative field name.
	<multi>	List of items. The field can contain multiple items separated by space.
	<item>...</item>	The structure is the same as described for <b>fixdic/fielddic/valblockdef/item</b> .
	</multi>	-
	<item>...</item>	Can be zero or more item elements in the field definition. Only one item can be specified for the field. The structure is the same as described for <b>fixdic/fielddic/valblockdef/item</b> .
	<range>...</range>	The structure is the same as described for <b>fixdic/fielddic/valblockdef/range</b> .
	<valblock idref="..."/>	Reference to the id of the <valblockdef> element.
	<descr>...</descr>	Description of the field.
	</fielddef>	-
	</fielddic>	-
	<msgdic>	Section which describes messages used in the dictionary.
	<blockdef>	Definition of the block. Can be zero or more definitions.
	@id	Identifier of the block definition.
	@name	Name of the block definition.
	@transport	Transport.
	<field>	Definition of the field.
	@tag	Field tag.
	@name	Field name.
	@req	Specified if the field is mandatory or not.
	@condreq	Specified if there is a condition when the field is mandatory.
	<comment>...</comment>	Comment.
	</field>	-
	<block>	Block description.
	@idref	Reference to the id of the block.
	@req	Specified if the block is mandatory or not.
	@condreq	Specified if there is a condition when the field is mandatory.
	<comment>...</comment>	Comment.
	</block>	-
	<group>	Group description.
	@nofield	Tag of the group.
	@startfield	Start field.
	<field>...</field>	The structure is the same as described for <b>fixdic/msgdic/blockdef/field</b> .
	<block>...</block>	The structure is the same as described for <b>fixdic/msgdic/blockdef/block</b> .
	<group>...</group>	The structure is the same as described for <b>fixdic/msgdic/blockdef/group</b> .
	</group>	-
	<descr>...</descr>	Description.
	</blockdef>	-
	<msgdef>	Definition of the message. Can be zero or more definitions.
	@msgtype	Message type.
	@name	Message name.
	@admin	Defines if the message is session level or application level message.
	<alias>...</alias>	Alternative message name. Can be zero or more elements.
	<field>...</field>	The structure is the same as described for <b>fixdic/msgdic/blockdef/field</b> .

<block>...</block>	The structure is the same as described for <i>fixdic/msgdic/blockdef/block</i> .
<group>...</group>	The structure is the same as described for <i>fixdic/msgdic/blockdef/group</i> .
<descr>...</descr>	Description.
</msgdef>	-
</msgdic>	-
<descr>...</descr>	Description.
</fixdic>	-

## Additional dictionary specifics

Below is the common structure of the additional dictionary with specific tags and attributes description.

Structure	Description
<fixdics>	Root tag of the additional dictionary.
<update>	Dictionary update operation. Can be zero or one <i>&lt;update&gt;</i> element.
<fixdic>	Dictionary to be updated. Can be zero or more <i>&lt;fixdic&gt;</i> elements.
@id	ID of the additional dictionary.
@parent_id	ID of the original dictionary additional dictionary based on. Valid values are: <ul style="list-style-type: none"> <li>• FIX40</li> <li>• FIX41</li> <li>• FIX42</li> <li>• FIX43</li> <li>• FIX44</li> <li>• FIX50</li> <li>• FIX50SP1</li> <li>• FIX50SP2</li> <li>• FIX50SP2EP</li> <li>• FIXT11</li> </ul>
@fixversion	FIX version additional dictionary based on.
@title	The name of the additional dictionary.
@version	Additional dictionary version.
@date	Date of the last additional dictionary update.
...	<i>&lt;typedef&gt;/&lt;valblockdef&gt;/&lt;fielddef&gt;/&lt;blockdef&gt;/&lt;msgdef&gt;</i> to be added/updated. The structure of listed elements is the same as described in standalone dictionary structure.
</fixdic>	-
</update>	-
</fixdics>	-

Please see the article [How to use custom dictionaries](#) for the cases of additional dictionaries usage.



There is no way to remove an elements from the base FIX dictionary

## How to manage list of dictionaries in FIXAntenna based applications

List of dictionaries to be used by FIXAntenna based applications can be managed using *DictionariesFilesList* property in `engine.properties` configuration file.

This parameter contains a list of names of XML files with definitions and extensions of the FIX protocols delimited by semicolon.

It may contain both standard and custom dictionaries:

```
DictionariesFilesList = ../../data/fixdic40.xml;../../data/fixdic41.xml;../../data/fixdic42.xml;../../data/fixdic43.xml;../../data/fixdic44.xml;../../data/fixdic50.xml;../../data/fixdic50sp1.xml;../../data/fixdic50sp2.xml;../../data/fixdict11.xml;../../data/additional_dict.xml
```

## Dictionary customization

As mentioned above, dictionaries can be customized for needs of particular exchange.

Refer to [How to use custom dictionaries with FIXEdge](#) article for details.

## How to check that dictionary doesn't have errors

[Simple Client](#) can be used in order to check that the dictionary doesn't have errors.

Below is step-by-step instruction how to check the dictionary with the help of [Simple Client](#):

1. Open the folder with SimpleClient;
2. Specify prepared dictionary in [DictionariesFilesList](#) property in `engine.properties` configuration file [Simple Client](#) refers on;
3. Run the Simple Client;
4. Check the results of launch:
  - a. If specified dictionary doesn't have errors, Simple Client will be launched successfully;
  - b. If specified dictionary has any error, Simple Client won't be started and the message with the reason will appear:

