

FIXML Dictionaries

Customization Guide

Table of Contents

Overview	3
How to customize a FIX dictionary	4
Fixdic.xml structure.....	4
Examples of different customizations	6
1. How to add custom value to existing FIX tag.....	6
2. How to change 'required' attribute of a tag in FIX message	8
3. How to change 'required' attribute of a tag in component block	9
4. How to change 'required' attribute of a component block in FIX message	10
5. How to add custom tag to existing FIX message.....	11
6. How to add custom tag to existing FIX component block	13
7. How to add custom repeating group to existing FIX message.....	14
8. How to add custom FIX message.....	15
How to customize a FIX dictionary	16
1. How to add custom value to existing FIX tag.....	17
2. How to change 'required' attribute of a tag in FIX message	18
3. How to change 'required' attribute of a tag in component block	19
4. How to change 'required' attribute of a component block in FIX message	20
5. How to add custom tag to existing FIX message.....	21
6. How to add custom tag to existing FIX component block	24
7. How to add custom repeating group to existing FIX message.....	26
8. How to add custom FIX message.....	28
Contact us	30

Overview

This document describes how to customize FIX and FIXML dictionaries

The document describes how FIX and FIXML dictionaries are built, how to customize dictionaries: how to define a custom tag; how to extend existing message with a custom tag; how to extend the list of custom tag values; how to define a custom message etc.

!Important: When you are applying customizations for FIX and FIXML dictionaries to build a custom FIX ↔ FIXML convertor make sure that both dictionaries fully correspond to each other.

How to customize a FIX dictionary

In order to apply any customizations for a standard FIX Dictionary it is necessary to do the following:

1. Refer to fixdic.xml
2. Apply changes
3. Store changes to the file

Fixdic.xml structure

FIX dictionary consists of the following blocks:

Block name	Description
<fixdic>	FIX dictionary main node
<typelist>	Data types list definition
<typedef >	Data type definition
<fielddic>	FIX tags and values identifier
<valblockdef>	FIX tag values block definition
<fielddef>	FIX tag definition
<msgdic>	Messages and component blocks definition
<blockdef>	Definition of the Component Block used in FIX message
<msgdef>	FIX message definition
<group>	Repeating group definition

An example of FIXdic.xml structure is given below.

```

<fixdic fixversion="5.0" title="FIX dictionary name (FIX 5.0)" date="2010/02/14">
  <typelist>
    <typedef type="int"/>
    ....
  </typelist>

  <fielddic>
    <valblockdef id="ValBlockID" name="ValBlockName">
      <item val="Val">Value Description</item>
    </valblockdef>
    .....
    <valblockdef/>

    <fielddef tag="TagID" name="TagName" type="DataType">
      <descr>
        <p>Descriptive text.</p>
      </descr>
    </fielddef>
    .....
    < fielddef/>
  </fielddic>

  <msgdic>
    <blockdef id="ComponentBlockID" name="ComponentBlockName">
      <field tag="TagID" name="TagName" />
    </blockdef>
    .....
    <blockdef/>

    <msgdef msgtype="MessageTypeID" name="MessageName">
      <alias>MessageAlias</alias>
      <field tag="TagID" name="TagName" />
      <block idref="ComponentBlockID" req="Y" />
      <field tag="386" name="LeadingTag"/>
      <group nofield="LeadingTagID" startfield="StartTagID">
        <field tag="StartTagID" name="StartTagName ">
          <field tag="GroupTagID" name="GroupTagName"/>
        </group>
      <descr>
        <p>Message description</p>
      </descr>
    </msgdef>
    .....
    <msgdef/>
  </msgdic>
</fixdic>

```

Examples of different customizations

1. How to add custom value to existing FIX tag

Example 1: Valid values for [AllocRejCode \(88\)](#) tag should be extended with custom value "14" (unknown clearing account).

Definition: [AllocRejCode \(88\)](#) tag is defined in the following way:

```
<fielddef tag="88" name="AllocRejCode" type="int">
  <valblock idref="AllocRejCode"/>
  <descr>
    <p>Identifies reason for rejection</p>
  </descr>
</fielddef>
```

According to this definition it is necessary to refer to FIX tags values blocks named "[AllocRejCode](#)" and specify the following (refer to text marked bold dark blue)

```
<valblockdef id="AllocRejCode" name="AllocRejCode">
  <item val="0">unknown account(s)</item>
  <item val="1">incorrect quantity</item>
  <item val="2">incorrect average price</item>
  .....
  <item val="12">unknown<fieldref tag="11">CIOrdID</fieldref></item>
  <item val="13">warehouse request rejected</item>
  <item val="14"> unknown clearing account </item>
</valblockdef>
```

Example 2: Valid values for **MDEntryType (269)** tag should be extended with custom value **"R"** (Rolling average).

Definition: Find **MDEntryType (269)** tag definition in the file and add a value as illustrated in the example below:

```
<fielddef tag="269" name="MDEntryType" type="char">
  <item val="0">Bid</item>
  <item val="1">Offer</item>
  <item val="2">Trade</item>
  ....
  <item val="B">Trade Volume</item>
  <item val="C">Open Interest</item>
  <item val="R">Rolling average</item>
  <descr>
    <p>Type Market Data entry.</p>
  </descr>
</fielddef>
```

2. How to change 'required' attribute of a tag in FIX message

Example: It is necessary to change the 'required' attribute for Password (554) tag from 'not required' to 'required' in FIX message User Request (MsgType = 'BE')

Definition: Refer to <msgdic> section and find User Request (MsgType = 'BE') message definition. Find Password (554) tag definition in message definition and add the following attribute (if absent) req = "Y" (refer to marked bold dark blue).

Note: If the tag was defined as 'not required' via req = "N" attribute, its value should be changed to "Y".

```
<msgdef msgtype="BE" name="User Request">
  <field tag="923" name="UserRequestID" req="Y"/>
  <field tag="924" name="UserRequestType" req="Y"/>
  <field tag="553" name="Username" req="Y"/>
  <field tag="554" name="Password" req = "Y"/>
  <field tag="925" name="NewPassword"/>
  <field tag="95" name="RawDataLength" condreq="existtags(T$96)"/>
  <field tag="96" name="RawData" condreq="T$95>0">
    <comment>Can be used to hand structures etc to other API's etc</comment>
  </field>
  <descr>
    <p>This message is used to initiate a user action, logon, logout or password change. It
    can also be used to request a report on a user's status.</p>
  </descr>
</msgdef>
```


3. How to change 'required' attribute of a tag in component block

Example: It is necessary to change the 'required' attribute for **Commission (12)** and **Commission Type (13)** tags from 'not required' to 'required' in **Commission Data** component block.

Definition: Refer to `<msgdic>` section and find **Commission Data** component block definition. Define the following attribute (if absent) `req = "Y"` in component block **Commission (12)** and **Commission Type (13)** tags (refer to marked bold dark blue).

Note: If the tag was defined as 'not required' via `req = "N"` attribute, its value should be changed to "Y".

```
<blockdef id="CommissionData" name="Commission Data">
  <field tag="12" name="Commission" req = "Y"/>
  <field tag="13" name="CommType" req = "Y"/>
  <field tag="479" name="CommCurrency"/>
  <field tag="497" name="FundRenewWaiv"/>
</blockdef>
```

4. How to change 'required' attribute of a component block in FIX message

Example: It is necessary to change 'required' attribute in Settlement Instruction Request (MsgType = 'AV') message for "Parties" component block from 'not required' to 'required'.

Definition: Refer to <msgdic> section and find Settlement Instruction Request (MsgType = 'AV') message definition. Find "Parties" component block definition in message definition and add the following attribute (if absent) req = "Y" (refer to marked bold dark blue).

Note: If the tag was defined as 'not required' via req = "N" attribute, its value should be changed to "Y".

```
<msgdef msgtype="AV" name="Settlement Instruction Request">
  <field tag="791" name="SettlInstReqID" req="Y"/>
  <field tag="60" name="TransactTime" req="Y"/>
  <block idref="Parties" req = "Y">
    <comment>Used here for party whose instructions this message is requesting and
      (optionally) for settlement location. Not required if database identifiers are being used to
      request settlement instructions. Required otherwise.</comment>
  </block>
  <field tag="79" name="AllocAccount"/>
  <field tag="661" name="AllocAcctIDSource"/>
  <field tag="54" name="Side"/>
  <field tag="460" name="Product"/>
  <field tag="167" name="SecurityType"/>
  <field tag="461" name="CFIcode"/>
  <field tag="168" name="EffectiveTime"/>
  <field tag="126" name="ExpireTime"/>
  <field tag="779" name="LastUpdateTime"/>
  <field tag="169" name="StandInstDbType"/>
  <field tag="170" name="StandInstDbName"/>
  <field tag="171" name="StandInstDbID" condreq="existstags(T$169)"/>
  <descr> ... </descr>
</msgdef>
```

Note: The same can be done if it is necessary to change the 'required' attribute for tag inside a repeating group.

5. How to add custom tag to existing FIX message

Example 1: It is necessary to add a custom tag **PolicyCode (8060)** as 'not required' field in **New Order Single (MsgType = 'D')** message. The tag has no predefined valid values; data type for values is String.

Definition:

1. Declare a custom FIX tag in fields dictionary. Refer to [<fielddic>](#) section and define the following

```
<fielddef tag="8060" name="PolicyCode" type="String"/>
  <descr>
    <p>Policy code description</p>
  </descr>
</fielddef>
```

Note: [<descr>](#) contains tag description, it is not required and can be omitted.

2. Refer to [<msgdic>](#) and find **New Order Single (MsgType = 'D')** description and add **PolicyCode (8060)** as described below in bold dark blue:

```
<msgdef msgtype="D" name="New Order - Single">
  <alias>New Order</alias>
  <alias>New Order Single</alias>
  <field tag="11" name="ClOrdID" req="Y"/>
  <block idref="Parties"/>
  <field tag="229" name="TradeOriginationDate"/>
  <field tag="75" name="TradeDate"/>
  <field tag="1" name="Account"/>
  .....
  <field tag="8060" name="PolicyCode"/>
  <descr> ... </descr>
</msgdef>
```

Example 2: It is necessary to add a custom tag **VTRejectReasonBid (8100)** as **required** field in **Mass Quote Acknowledgement (MsgType = 'b')** message. The tag has the following predefined valid values:

- "0" – Exchange option
- "1" – Unknown or non-tradable contract
- "2" – Exchange closed
- "99" – Other

Definition:

1. Declare a custom FIX tag in fields dictionary. Refer to [<fielddic>](#) section and define the following

```
<fielddef tag="8100" name="VTRejectReasonBid" type="int">
  <item val="0">Exchange option</item>
  <item val="1">Unknown or non-tradable contract</item>
  <item val="2">Exchange closed</item>
  <item val="99">Other</item>
</fielddef>
```

2. Refer to [<msgdic>](#) and find **Mass Quote Acknowledgement (MsgType = 'b')** message description and add **VTRejectReasonBid (8100)** as described below in bold dark blue:

```
<msgdef msgtype="b" name="Mass Quote Acknowledgement">
  <alias>MassQuotAck</alias>
  <field tag="131" name="QuoteReqID"/>
  <field tag="117" name="QuoteID"/>
  <field tag="297" name="QuoteStatus"/>
  <field tag="300" name="QuoteRejectReason"/>
  ....
  <field tag="8100" name="VTRejectReasonBid" req = "Y"/>
  ....
  <descr> ... </descr>
</msgdef>
```

6. How to add custom tag to existing FIX component block

Example: It is necessary to add a custom tag *ShareTier (8800)* as **not required** field in 'Commission' component block. The tag has the following predefined valid values: 'A', 'B', 'C'; data type for values is char.

Definition:

1. Declare a custom FIX tag in fields dictionary. Refer to [<fielddic>](#) section and define the following

```
<fielddef tag="8800" name=" ShareTier " type="char">
  <item val="A">Tier A</item>
  <item val="B">Tier B</item>
  <item val="C">Tier C</item>
  <descr> <p>Share Tier</p></descr>
</fielddef>
```

Note: [<descr>](#) contains tag description, it is not required and can be omitted.

2. Refer to [<msgdic>](#), find 'Commission' component block description and add *ShareTier (8800)* as described below in bold dark blue:

```
<blockdef id="CommissionData" name="Commission Data">
  <field tag="12" name="Commission" />
  <field tag="13" name="CommType" />
  <field tag="8800" name="ShareTier" />
  <field tag="479" name="CommCurrency"/>
  <field tag="497" name="FundRenewWaiv"/>
</blockdef>
```

7. How to add custom repeating group to existing FIX message

Example: It is necessary to add a custom repeating group in [List Execute \(MsgType = 'L'\)](#) message. The repeating group consists of the following fields:

- leading tag: [NoTradingSessions \(386\)](#)
- repeating group fields:
 - start field: [TradingSessionID \(336\)](#)
 - other fields: [TradingSessionSubID \(625\)](#)

The repeating group is not required in message.

Definition: Refer to [<msgdic>](#) , find [List Execute \(MsgType = 'L'\)](#) message description and add a repeating group as described below in bold dark blue:

```
<msgdef msgtype="L" name="List Execute">
  <alias>ListExct</alias>
  <field tag="66" name="ListID" req="Y"/>
  <field tag="391" name="ClientBidID"/>
  <field tag="390" name="BidID"/>
  <field tag="60" name="TransactTime" req="Y"/>
  <field tag="58" name="Text"/>
  <field tag="354" name="EncodedTextLen" condreq="existtags(T$355)"/>
  <field tag="355" name="EncodedText"/>
  <field tag="386" name="NoTradingSessions">
    <comment>Specifies the number of repeating TradingSessionIDs</comment>
  </field>
  <group nofield="386" startfield="336">
    <field tag="336" name="TradingSessionID">
      <comment>Required if<fieldref tag="386">NoTradingSessions</fieldref>is > 0.
    </comment>
    </field>
    <field tag="625" name="TradingSessionSubID"/>
  </group>
  <descr> ... </descr>
</msgdef>
```

Note: The custom repeating group can be added to component block or into another repeating group in the same way.

8. How to add custom FIX message

Example: It is necessary to add a custom [Start One Side Only Period Confirmation](#) (MsgType = 'U0505') message. The message consists of the following fields:

- [ClOrdID](#) (11) – required
- [Side](#) (54) – required
- [Text](#) (58) – not required
- [TransactTime](#) (60) – required

Message description: A Start One Side Only Period Confirmation message is sent by the Euronext to confirm the reception of the Start One Side Only Period. This message is only available for Euronext markets.

The repeating group is not required in message.

Definition: Refer to `<msgdic>` and add [Start One Side Only Period Confirmation](#) (MsgType = 'U0505') message as described below:

```
<msgdef msgtype="U0505" name="Start One Side Only Period Confirmation">
  <alias>StartOneSideOnlyPeriodConfirmation</alias>
  <field tag="11" name="ClOrdID" req="Y" />
  <block idref="Instrument" req="Y" />
  <field tag="54" name="Side" req="Y" />
  <field tag="58" name="Text" />
  <field tag="60" name="TransactTime" req="Y" />
  <descr>
    <p>A Start One Side Only Period Confirmation message is sent by the Euronext to confirm
    the reception of the Start One Side Only Period. This message is only available for Euronext
    markets.</p>
  </descr>
</msgdef>
```

How to customize a FIX dictionary

In order to apply any customizations for standard FIXML Dictionary, it is necessary to do the following:

4. Refer to fixml schema
5. Apply changes
6. Store changes to the file

Layers of FIXML schema:

Layer	Description
fixml-datatypes-*-*.xsd	FIXML data types definition
fixml-fields-base-*-*.xsd fixml-fields-impl-*-*.xsd	FIXML fields and enumerations definitions
Fixml-components-base-*-*.xsd	FIXML root elements and global component blocks definition
fixml-categoryName-*-*.xsd	FIXML messages and components for each message category in FIXML definition

- *-* refers to FIX protocol version
- categoryName refers to FIXML category (per-trade; trade; order; etc)
E.g. fixml-order-4-4.xsd refers to all messages for Single Order workflow in FIX 4.4 Call Action

For detailed information refer to:

- All standard FIXML schemas at: <http://www.fixprotocol.org/specifications/>
- FIXML Technical documentation: <http://www.fixprotocol.org/specifications/TechDoc-FIXML>

All examples below are provided based on FIX 4.4.

1. How to add custom value to existing FIX tag

Example 1: Valid values for `AllocRejCode` (88) tag should be extended with custom value "14" (unknown clearing account).

Definition: Refer to `fixml-fields-base-4-4.xsd` and apply a custom value for `AllocRejCode` (88) in the following way (refer to bold dark blue):

```
<xs:simpleType name="AllocRejCode_enum_t">
  <xs:annotation>
    <xs:documentation xml:lang="en">..... </xs:documentation>
    <xs:appinfo xmlns:x="http://www.fixprotocol.org/fixml/metadata.xsd">
      <xs:Xref Protocol="FIX" name="AllocRejCode" tag="88" datatype="int"
        ComponentType="Field" StdAbbrev="RejCode" QualifiedAbbrev="RejCode"
        Category="**BaseCatagory**"
        CategoryAbbrev="**BaseCatagoryXMLName**" />
      <xs:Xref Protocol="ISO_15022_XML"/>
    </xs:appinfo>
    <xs:appinfo xmlns:x="http://www.fixprotocol.org/fixml/metadata.xsd">
      <x:EnumDoc value="0" desc="UnknownAcct" />
      <x:EnumDoc value="1" desc="IncorrectQty" />
      <x:EnumDoc value="2" desc="IncorrectAvgPrc" />
      <x:EnumDoc value="12" desc="unknownCIOrdID" />
      <x:EnumDoc value="13" desc="warehouseRequestRejected" />
      <x:EnumDoc value="14" desc="unknownClearingAccount" />
    </xs:appinfo>
  </xs:annotation>
  <xs:restriction base="xs:string">
    <xs:enumeration value = "0"/>
    <xs:enumeration value = "1"/>
    <xs:enumeration value = "2"/>
    ....
    <xs:enumeration value = "12"/>
    <xs:enumeration value = "13"/>
    <xs:enumeration value = "14"/>
  </xs:restriction>
</xs:simpleType>
```

2. How to change 'required' attribute of a tag in FIX message

Example: It is necessary to change 'required' attribute for Password (554) tag in FIX message User Request (MsgType = 'BE') from 'not required' to 'required'

Definition: Refer to [fixml-components-base-4-4.xsd](#) and find User Request (MsgType = 'BE') message definition. Find "Password" tag definition in message definition and change the attribute 'use' (if not absent) to required as **use = "required"** (refer to marked bold dark blue).

```
<xs:group name="UserRequestElements">
  <xs:sequence/>
</xs:group>
<xs:attributeGroup name="UserRequestAttributes">
  <xs:attribute name="UserReqID" type="UserRequestID_t" use="required"/>
  <xs:attribute name="UserReqTyp" type="UserRequestType_t" use="required"/>
  <xs:attribute name="Username" type="Username_t" use="required"/>
  <xs:attribute name="Password" type="Password_t" use="required"/>
  <xs:attribute name="NewPassword" type="NewPassword_t" use="optional"/>
  <xs:attribute name="RawDataLength" type="RawDataLength_t" use="optional"/>
  <xs:attribute name="RawData" type="RawData_t" use="optional"/>
</xs:attributeGroup>

<xs:complexType name="UserRequest_message_t" final="#all">
  <xs:annotation>
    <xs:documentation xml:lang="en">UserRequest can be found in Volume1 of the specification
  </xs:documentation>
    <xs:appinfo xmlns:x="http://www.fixprotocol.org/fixml/metadata.xsd">
      <xs:Xref Protocol="FIX" name="UserRequest" ComponentType="Message"/>
      <xs:Xref Protocol="ISO_15022_XML"/>
    </xs:appinfo>
  </xs:annotation>
  <xs:complexContent>
    <xs:extension base="Abstract_message_t">
      <xs:sequence> <xs:group ref="UserRequestElements"/> </xs:sequence>
      <xs:attributeGroup ref="UserRequestAttributes"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:element name="UserReq" type="UserRequest_message_t" substitutionGroup="Message" final="#all"/>
```

3. How to change 'required' attribute of a tag in component block

Example: It is necessary to change 'required' attribute for [Commission \(12\)](#) and [Commission Type \(13\)](#) tags in [Commission Data](#) component block from 'not required' to 'required'.

Definition: Refer to [fixml-components-base-4-4.xsd](#) section and find [Commission Data](#) component block definition. Refer to [Commission \(12\)](#) and [Commission Type \(13\)](#) tags definition and change attribute 'use' (if not absent) to required as use = "required" (refer to marked bold dark blue).

```
<xs:group name="CommissionDataElements">
  <xs:sequence/>
</xs:group>
<xs:attributeGroup name="CommissionDataAttributes">
  <xs:attribute name="Comm" type="Commission_t" use="required"/>
  <xs:attribute name="CommTyp" type="CommType_t" use="required"/>
  <xs:attribute name="Ccy" type="CommCurrency_t" use="optional"/>
  <xs:attribute name="FundRenewWaiv" type="FundRenewWaiv_t" use="optional"/>
</xs:attributeGroup>
<xs:complexType name="CommissionData_Block_t" final="#all">
  <xs:annotation>
    <xs:documentation xml:lang="en">CommissionData is a component
    </xs:documentation>
    <xs:appinfo xmlns:x="http://www.fixprotocol.org/fixml/metadata.xsd">
      <xs:Xref Protocol="FIX" name="CommissionData" ComponentType="Block"/>
      <xs:Xref Protocol="ISO_15022_XML"/>
    </xs:appinfo>
  </xs:annotation>
  <xs:sequence>
    <xs:group ref="CommissionDataElements"/>
  </xs:sequence>
  <xs:attributeGroup ref="CommissionDataAttributes"/>
</xs:complexType>
```

4. How to change 'required' attribute of a component block in FIX message

Example: It is necessary to change 'required' attribute of "Parties" component block in FIX message Settlement Instruction Request (MsgType = 'AV') from 'not required' to 'required'.

Definition: Refer to [fixml-settlement-base-4-4.xsd](#) and find Settlement Instruction Request (MsgType = 'AV') message definition. Find "Parties" component block definition in message definition and change the 'minOccurs' attribute to **minOccurs="1"** (refer to marked bold dark blue).

```
<xs:group name="SettlementInstructionRequestElements">
  <xs:sequence>
    <xs:element name="Pty" type="Parties_Block_t" minOccurs="1" />
  </xs:sequence>
</xs:group>
<xs:attributeGroup name="SettlementInstructionRequestAttributes">
  ....
</xs:attributeGroup>

<xs:complexType name="SettlementInstructionRequest_message_t" final="#all">
  <xs:annotation>
    <xs:documentation xml:lang="en">SettlementInstructionRequest can be found in Volume5 of
      the specification
    </xs:documentation>
    <xs:appinfo xmlns:x="http://www.fixprotocol.org/fixml/metadata.xsd">
      <xs:Xref Protocol="FIX" name="SettlementInstructionRequest"
        ComponentType="Message"/>
      <xs:Xref Protocol="ISO_15022_XML"/>
    </xs:appinfo>
  </xs:annotation>
  <xs:complexContent>
    <xs:extension base="Abstract_message_t">
      <xs:sequence>
        <xs:group ref="SettlementInstructionRequestElements"/>
      </xs:sequence>
      <xs:attributeGroup ref="SettlementInstructionRequestAttributes"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:element name="SettlInstrctnReq" type="SettlementInstructionRequest_message_t"
  substitutionGroup="Message" final="#all"/>
```

5. How to add custom tag to existing FIX message

Example 1: It is necessary to add custom tag **PolicyCode (8060)** as not required field in **New Order Single (MsgType = 'D')** message. The tag has no predefined valid values; data type for values is **String**.

Definition:

1. Declare a custom FIX tag in fields dictionary. Refer to [fixml-fields-base-4-4.xsd](#) section and define the following

```
<xs:simpleType name="PolicyCode_t">
  <xs:annotation>
    <xs:documentation xml:lang="en">PolicyCode in case of Best Execution.
    </xs:documentation>
    <xs:appinfo xmlns:x="http://www.fixprotocol.org/fixml/metadata.xsd">
      <xs:Xref Protocol="FIX" name="PolicyCode" tag="8060" datatype="String"
        ComponentType="Field" StdAbbrev="PolicyCod"
        QualifiedAbbrev="PolicyCod" Category="" CategoryAbbrev="" />
      <xs:Xref Protocol="ISO_15022_XML"/>
    </xs:appinfo>
  </xs:annotation>
  <xs:restriction base="xs:string"/>
</xs:simpleType>
```

2. Refer to [fixml-order-base-4-4.xsd](#) , find **New Order Single (MsgType = 'D')** description and add **PolicyCode (8060)** as described below in bold dark blue:

```
<xs:group name="NewOrderSingleElements">
  <xs:sequence>
    <xs:element name="Pty" type="Parties_Block_t" minOccurs="0" />
    <xs:element name="OrdQty" type="OrderQtyData_Block_t" />
    ....
  </xs:sequence>
</xs:group>
<xs:attributeGroup name="NewOrderSingleAttributes">
  <xs:attribute name="ID" type="CIOrdID_t" use="required"/>
  <xs:attribute name="TrdDt" type="TradeDate_t" use="optional"/>
  <xs:attribute name="Acct" type="Account_t" use="optional"/>
  ....
  <xs:attribute name="PolicyCode" type="PolicyCode_t" use="optional"/>
</xs:attributeGroup>
<xs:complexType name="NewOrderSingle_message_t" final="#all">
  <xs:annotation>
    <xs:documentation xml:lang="en">NewOrderSingle can be found in Volume4 of the
      specification</xs:documentation>
    <xs:appinfo xmlns:x="http://www.fixprotocol.org/fixml/metadata.xsd">
      <xs:Xref Protocol="FIX" name="NewOrderSingle" ComponentType="Message"/>
    </xs:appinfo>
  </xs:annotation>
</xs:complexType>
```

```

        <xs:Xref Protocol="ISO_15022_XML"/>
    </xs:appinfo>
</xs:annotation>
<xs:complexContent>
    <xs:extension base="Abstract_message_t">
        <xs:sequence>
            <xs:group ref="NewOrderSingleElements"/>
        </xs:sequence>
        <xs:attributeGroup ref="NewOrderSingleAttributes"/>
    </xs:extension>
</xs:complexContent>
</xs:complexType>

<xs:element name="Order" type="NewOrderSingle_message_t" substitutionGroup="Message" final="#all"/>
    
```

Example 2: It is necessary to add a custom tag `VTRejectReasonBid` (8100) as required field in `Mass Quote Acknowledgement` (`MsgType = 'b'`) message. The tag has the following predefined valid values:

- “0” – Exchange option
- “1” – Unknown or non-tradable contract
- “2” – Exchange closed
- “99” – Other

Definition:

1. Declare a custom FIX tag in fields dictionary. Refer to `fixml-fields-base-4-4.xsd` section and define the following

```

<xs:simpleType name="VTRejectReasonBid_enum_t">
    <xs:annotation>
        <xs:documentation xml:lang="en">If the Quote has been rejected an error code will be returned in this field.</xs:documentation>
    </xs:annotation>
    <xs:appinfo>
        <xs:Xref Protocol="FIX" name="VTRejectReasonBid" tag="8100" datatype="String" ComponentType="Field" StdAbbrev="VTRejRsnBid" QualifiedAbbrev="VTRejRsnBid" Category="" CategoryAbbrev="" />
        <xs:Xref Protocol="ISO_15022_XML"/>
    </xs:appinfo>
    <xs:appinfo xmlns:x="http://www.fixprotocol.org/fixml/metadata.xsd">
        <x:EnumDoc value="0" desc="Exchange option"/>
        <x:EnumDoc value="1" desc="Unknown or non-tradable contract"/>
        <x:EnumDoc value="2" desc="Exchange closed"/>
        <x:EnumDoc value="99" desc="Other"/>
    </xs:appinfo>
</xs:annotation>
<xs:restriction base="xs:string">
    <xs:enumeration value = "0"/>
    <xs:enumeration value = "1"/>
    <xs:enumeration value = "2"/>
    <xs:enumeration value = "99"/>
</xs:restriction>
</xs:simpleType>
    
```

2. Refer to [fixml-fields-impl-4-4.xsd](#) and declare:

```
<xs:simpleType name="VTRejectReasonBid_t">
  <xs:restriction base="VTRejectReasonBid_enum_t"/>
</xs:simpleType>
```

3. Refer to [fixml-quotation-base-4-4.xsd](#), find [Mass Quote Acknowledgement \(MsgType = 'b'\)](#) message description and add [VTRejectReasonBid \(8100\)](#) as described below in bold dark blue:

```
<xs:group name="MassQuoteAcknowledgementElements">
  <xs:sequence>
    <xs:element name="Pty" type="Parties_Block_t" minOccurs="0" />
    <xs:element name="QuotSetAck" type="QuotSetAckGrp_Block_t" minOccurs="0"
      maxOccurs="unbounded"/>
  </xs:sequence>
</xs:group>
<xs:attributeGroup name="MassQuoteAcknowledgementAttributes">
  <xs:attribute name="ReqID" type="QuoteReqID_t" use="optional"/>
  <xs:attribute name="QID" type="QuoteID_t" use="optional"/>
  <xs:attribute name="Stat" type="QuoteStatus_t" use="required"/>
  <xs:attribute name="RejRsn" type="QuoteRejectReason_t" use="optional"/>
  <xs:attribute name="VTRejectReasonBid" type="VTRejectReasonBid_t" use="required"/>
</xs:attributeGroup>

<xs:complexType name="MassQuoteAcknowledgement_message_t" final="#all">
  <xs:annotation>
    <xs:documentation xml:lang="en">MassQuoteAcknowledgement can be found in Volume3 of
      the specification
    </xs:documentation>
    <xs:appinfo xmlns:x="http://www.fixprotocol.org/fixml/metadata.xsd">
      <xs:Xref Protocol="FIX" name="MassQuoteAcknowledgement"
        ComponentType="Message"/>
      <xs:Xref Protocol="ISO_15022_XML"/>
    </xs:appinfo>
  </xs:annotation>
  <xs:complexContent>
    <xs:extension base="Abstract_message_t">
      <xs:sequence>
        <xs:group ref="MassQuoteAcknowledgementElements"/>
      </xs:sequence>
      <xs:attributeGroup ref="MassQuoteAcknowledgementAttributes"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

6. How to add custom tag to existing FIX component block

Example: It is necessary to add a custom tag **ShareTier (8800)** as not required field in 'Commission' component block. The tag has the following predefined valid values: 'A', 'B', 'C'; data type for values is char.

Definition:

1. Declare a custom FIX tag in fields dictionary. Refer to [fixml-fields-base-4-4.xsd](#) section and define the following

```
<xs:simpleType name=" ShareTier_enum_t">
  <xs:annotation>
    <xs:documentation xml:lang="en">Indicates Share Tier.</xs:documentation>
    <xs:appinfo xmlns:x="http://www.fixprotocol.org/fixml/metadata.xsd">
      <xs:Xref Protocol="FIX" name=" ShareTier " tag="8800" datatype="char"
        ComponentType="Field" StdAbbrev=" ShareTier" QualifiedAbbrev="
        ShareTier" Category="**BaseCatagory**"
        CategoryAbbrev="**BaseCatagoryXMLName**" />
      <xs:Xref Protocol="ISO_15022_XML"/>
    </xs:appinfo>
    <xs:appinfo xmlns:x="http://www.fixprotocol.org/fixml/metadata.xsd">
      <x:EnumDoc value="A" desc=" Tier A " />
      <x:EnumDoc value="B" desc=" Tier B" />
      <x:EnumDoc value="C" desc=" Tier C" />
    </xs:appinfo>
  </xs:annotation>
  <xs:restriction base="xs:int">
    <xs:enumeration value = "A"/>
    <xs:enumeration value = "B"/>
    <xs:enumeration value = "C"/>
  </xs:restriction>
</xs:simpleType>
```

2. Refer to [fixml-fields-impl-4-4.xsd](#) and declare:

```
<xs:simpleType name=" ShareTier_t">
  <xs:restriction base=" ShareTier_enum_t"/>
</xs:simpleType>
```

3. Refer to [fixml-components-base-4-4.xsd](#), find 'Commission' component block description and add **ShareTier (8800)** as described below in bold dark blue:

```
<xs:group name="CommissionDataElements">
  <xs:sequence/>
</xs:group>
<xs:attributeGroup name="CommissionDataAttributes">
  <xs:attribute name="Comm" type="Commission_t" use="optional"/>
  <xs:attribute name="CommTyp" type="CommType_t" use="optional"/>
  <xs:attribute name=" ShareTier" type=" ShareTier_t" use="optional"/>
</xs:attributeGroup>
```



```
<xs:attribute name="Ccy" type="CommCurrency_t" use="optional"/>
<xs:attribute name="FundRenewWaiv" type="FundRenewWaiv_t" use="optional"/>
</xs:attributeGroup>
<xs:complexType name="CommissionData_Block_t" final="#all">
  <xs:annotation>
    <xs:documentation xml:lang="en">CommissionData is a component
    </xs:documentation>
    <xs:appinfo xmlns:x="http://www.fixprotocol.org/fixml/metadata.xsd">
      <xs:Xref Protocol="FIX" name="CommissionData" ComponentType="Block"/>
      <xs:Xref Protocol="ISO_15022_XML"/>
    </xs:appinfo>
  </xs:annotation>
  <xs:sequence>
    <xs:group ref="CommissionDataElements"/>
  </xs:sequence>
  <xs:attributeGroup ref="CommissionDataAttributes"/>
</xs:complexType>
```

7. How to add custom repeating group to existing FIX message

Example: It is necessary to add a custom repeating group in [List Execute \(MsgType = 'L'\)](#) message. The repeating group consists of the following fields:

- leading tag: [NoTradingSessions \(386\)](#)
- repeating group fields:
 - start field: [TradingSessionID \(336\)](#)
 - other fields: [TradingSessionSubID \(625\)](#)

The repeating group is not required in message.

Definition:

1. Make sure that a repeating group is defined in the schema. 'Trading Sessions' is a standard FIX repeating group which is defined in [fixml-components-base-4-4.xsd](#) as illustrated in the example below:

```
<xs:group name="TrdgSesGrpElements">
  <xs:sequence/>
</xs:group>
<xs:attributeGroup name="TrdgSesGrpAttributes">
  <xs:attribute name="SesID" type="TradingSessionID_t" use="optional"/>
  <xs:attribute name="SesSub" type="TradingSessionSubID_t" use="optional"/>
</xs:attributeGroup>
<xs:complexType name="TrdgSesGrp_Block_t" final="#all">
  <xs:annotation>
    <xs:documentation xml:lang="en">TrdgSesGrp is a component
  </xs:documentation>
    <xs:appinfo xmlns:x="http://www.fixprotocol.org/fixml/metadata.xsd">
      <xs:Xref Protocol="FIX" name="TrdgSesGrp"
        ComponentType="ImplicitBlockRepeating"/>
      <xs:Xref Protocol="ISO_15022_XML"/>
    </xs:appinfo>
  </xs:annotation>
  <xs:sequence>
    <xs:group ref="TrdgSesGrpElements"/>
  </xs:sequence>
  <xs:attributeGroup ref="TrdgSesGrpAttributes"/>
</xs:complexType>
```

2. Refer to [fixml-listorders-base-4-4.xsd](#), find [List Execute \(MsgType = 'L'\)](#) message description and add the repeating group as described below in bold dark blue:

```
<xs:group name="ListExecuteElements">
  <xs:sequence>
    <xs:element name="TrdSes" type="TrdgSesGrp_Block_t" minOccurs="0"
      maxOccurs="unbounded"/>
  </xs:sequence>
</xs:group>
<xs:attributeGroup name="ListExecuteAttributes">
  <xs:attribute name="ID" type="ListID_t" use="required"/>
  <xs:attribute name="CIBidID" type="ClientBidID_t" use="optional"/>
</xs:attributeGroup>
```

```

<xs:attribute name="BidID" type="BidID_t" use="optional"/>
<xs:attribute name="TxnTm" type="TransactTime_t" use="required"/>
<xs:attribute name="Txt" type="Text_t" use="optional"/>
<xs:attribute name="EncTxtLen" type="EncodedTextLen_t" use="optional"/>
<xs:attribute name="EncTxt" type="EncodedText_t" use="optional"/>
</xs:attributeGroup>

<xs:complexType name="ListExecute_message_t" final="#all">
  <xs:annotation>
    <xs:documentation xml:lang="en">ListExecute can be found in Volume4 of the specification
    </xs:documentation>
    <xs:appinfo xmlns:x="http://www.fixprotocol.org/fixml/metadata.xsd">
      <xs:Xref Protocol="FIX" name="ListExecute" ComponentType="Message"/>
      <xs:Xref Protocol="ISO_15022_XML"/>
    </xs:appinfo>
  </xs:annotation>
  <xs:complexContent>
    <xs:extension base="Abstract_message_t">
      <xs:sequence>
        <xs:group ref="ListExecuteElements"/>
      </xs:sequence>
      <xs:attributeGroup ref="ListExecuteAttributes"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<xs:element name="ListExct" type="ListExecute_message_t" substitutionGroup="Message" final="#all"/>

```

Note: The custom repeating group can be added to component block or into another repeating group in the same way.

8. How to add custom FIX message

Example: It is necessary to add a custom [Start One Side Only Period Confirmation](#) (MsgType = 'U0505') message. The message consists of the following fields:

- [ClOrdID \(11\)](#) - required
- [Side \(54\)](#) – required
- [Text \(58\)](#) – not required
- [TransactTime \(60\)](#) – required

Message description: A Start One Side Only Period Confirmation message is sent by the Euronext to confirm the reception of the Start One Side Only Period. This message is only available for Euronext markets.

The repeating group is not required in message.

Important: When customizations for FIX and FIXML dictionaries are provided to build a custom FIX ↔ FIXML convertor be aware of the following:

While you are declaring a custom FIX message in FIXML dictionary make sure that FIX message alias (in FIX dictionary) is equal to FIXML message name defined as: ' <xs:element name="StrtOneSideOnlyPrdCnfrm" ...'

Definition:

1. Add user defined message to the list of FIXML categories.

Refer to [fixml-main-4-4.xsd](#) and specify the following (marked bold dark blue):

```
<xs:schema targetNamespace="http://www.fixprotocol.org/FIXML-4-4"
xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns="http://www.fixprotocol.org/FIXML-4-4"
elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xs:include schemaLocation="fixml-pretrade-4-4.xsd"/>
  <xs:include schemaLocation="fixml-trade-4-4.xsd"/>
  <xs:include schemaLocation="fixml-posttrade-4-4.xsd"/>
  <xs:include schemaLocation="fixml-userdefined-base-4-4.xsd"/>
</xs:schema>
```

2. Create the files:

- a. [fixml-userdefined-4-4.xsd](#)

With the following content:

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema targetNamespace="http://www.fixprotocol.org/FIXML-4-4"
xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns="http://www.fixprotocol.org/FIXML-4-4"
elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xs:include schemaLocation="fixml-userdefined-impl-4-4.xsd"/>
</xs:schema>
```

b. fixml-userdefined-impl-4-4.xsd

With the following content:

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema targetNamespace="http://www.fixprotocol.org/FIXML-4-4"
xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns="http://www.fixprotocol.org/FIXML-4-4"
elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xs:include schemaLocation="fixml-userdefined-base-4-4.xsd"/>
</xs:schema>
```

c. fixml-userdefined-base-4-4.xsd - in this file **Start One Side Only Period Confirmation (MsgType = 'U0505')** message should be defined as given below:

```
<?xml version="1.0" encoding="UTF-8"?>
  <xs:group name="StartOneSideOnlyPeriodConfirmationElements">
    <xs:sequence>
      <xs:element name="Instrmt" type="Instrument_Block_t" />
    </xs:sequence>
  </xs:group>
  <xs:attributeGroup name="StartOneSideOnlyPeriodConfirmationAttributes">
    <xs:attribute name="ID" type="ClOrdID_t" use="required"/>
    <xs:attribute name="Side" type="Side_t" use="required"/>
    <xs:attribute name="TxnTm" type="TransactTime_t" use="required"/>
    <xs:attribute name="Txt" type="Text_t" use="optional"/>
  </xs:attributeGroup>
  <xs:complexType name="StartOneSideOnlyPeriodConfirmation_message_t" final="#all">
    <xs:annotation>
      <xs:documentation xml:lang="en">A Start One Side Only Period message can be sent by the
        Liquidity Provider per instrument to indicate the start of a period in which the LP will
        only send buy or sell quotes for a specific instrument. To end the One Side Only
        period, the Liquidity Provider sends a quote on the opposite side of the orderbook.
        This message is only available for Euronext markets.
      </xs:documentation>
      <xs:appinfo xmlns:x="http://www.fixprotocol.org/fixml/metadata.xsd">
        <xs:Xref Protocol="FIX" name="ChangeReferencePriceResponse"
          ComponentType="Message"/>
        <xs:Xref Protocol="ISO_15022_XML"/>
      </xs:appinfo>
    </xs:annotation>
    <xs:complexContent>
      <xs:extension base="Abstract_message_t">
        <xs:sequence>
          <xs:group ref="StartOneSideOnlyPeriodConfirmationElements"/>
        </xs:sequence>
        <xs:attributeGroup ref="StartOneSideOnlyPeriodConfirmationAttributes"/>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:element name="StrtOneSideOnlyPrdCnfrm" type="StartOneSideOnlyPeriodConfirmation_message_t"
substitutionGroup="Message" final="#all"/>
```

Contact us

sales@btobits.com

Phone: +1-888-378-0666

Global Headquarters

US Client Support and Delivery Center

EPAM Systems, Inc

41 University Drive

Suite 202

Newtown, PA 18940

Phone: +1-267-759-9000

Fax: +1-267-759-8989