

# MDP 3.0 - Message Schema

This topic describes the CME Group MDP 3.0 message schema. CME Group provides a single XML schema file that contains FIX templates for all message types sent by CME Globex. A message schema is composed of metadata that specifies the components of a message and their data types and identifiers. For Simple Binary Encoding, message schemas are expressed as an XML template.

A FIX template corresponds to a FIX message type, and uniquely identifies an ordered collection of fields. The template also includes syntax indicating the type of field and transfer decoding to apply. A template is communicated between CME Group and client systems in XML syntax using the Simple Binary Encoding XML schema. The XML format is human- and machine-readable and can be used for authoring and storing FIX templates. Session Control Protocol (SCP) will not be used.

- [Schema Distribution](#)
- [Schema Versioning](#)
  - [Word Boundaries / Block Alignment](#)
  - [SBE Optimization](#)
- [Template Extension](#)
  - [Example: Template Before Appended Extension](#)
  - [Example: Template After Appended Extension](#)
  - [Non-Appended Extension](#)
- [Schema Format](#)

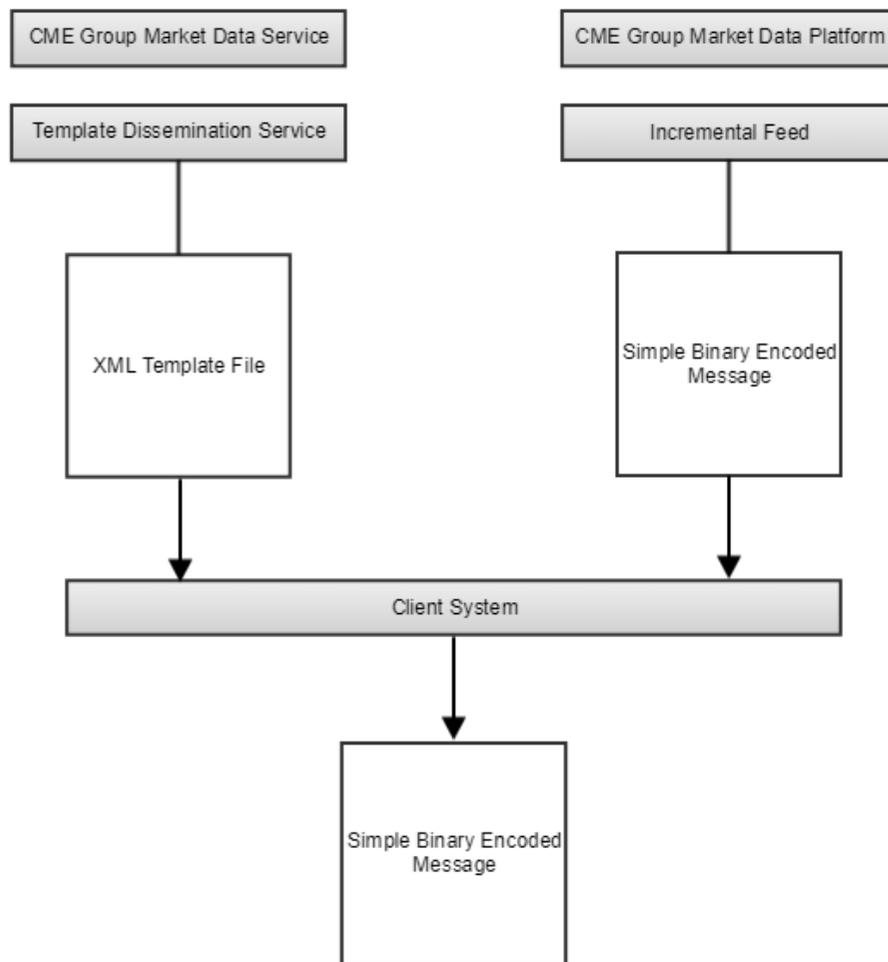
Each template has a unique Template ID that describes the format of the binary encoded message. A Template ID is present in every message to provide a reference to the correct template. The Template ID is unique and is included in every message header, allowing the client system to apply the correct schema to the message upon receiving it.

Client systems do not have to decode messages. They can choose to interpret the message directly in its binary format using the schema/template files.



Multiple messages may be included in a packet, so multiple Template IDs may be included in a packet.

The following diagram shows how the CME Group Market Data service provides the message templates required by the client system to process the Incremental market data feed.



## Schema Distribution

Schema and configuration files are available for download via the CME Group FTP (<https://www.cmegroup.com/ftp>) or SFTP (<sftpng.cmegroup.com>) site.

## Schema Versioning

The templates.xml file is versioned each time an update is made. All elements in a message schema are of the same version. The first version of a schema is version zero, and the version number is incremented each time a schema is changed. The “sinceVersion” attribute indicates when a template is extended.

## Word Boundaries / Block Alignment

To aid client system processing of MDP data, the root lengths of the most common messages—such as the Market Data Incremental Refresh (tag 35-MsgType=X), Market Data Security State (tag 35-MsgType=f), and the Request for Quote (tag 35-MsgType = R), that are not in increments of 8 bytes—are padded to the byte boundary in multiples of 8 byte units. Additionally, repeating block length is 8 byte aligned in the Incremental Messages.

## SBE Optimization

SBE has two features to control alignment of message elements.

- The length of a block is controlled with the **blockLength** attribute of a message or group. When applied to a message, it controls the length of the root level of the message, prior to any repeating groups or variable-length data. When applied to a group, it controls the length of each entry of the repeating group. The blockLength attribute is not required. By default, the length of a block is the sum of its field sizes. When specified, it must be at least that much, but can be greater. When greater than the field lengths, the extra space resides at the end of the block.
- The position of an individual field can be controlled with the **offset** attribute. Since individual field alignment is not considered crucial, this field attribute is optional. When not specified, the field is packed to the previous field without padding.

## Template Extension

In the event that a message template is updated, the change can be made as an *appended* extension, which requires no immediate action on the part of the client, or a *non-appended* extension, which requires implementation of the new template.

Template extensions can be applied to the body and/or repeating groups of an SBE message. When a template is extended as an appended extension, customers can choose to continue using the prior template version or process the new data with the new schema version.



Customers who support template extension should verify any SBE schema update in New Release prior to a Production schema launch to ensure compatibility.

For the **body** and **repeating groups** of SBE messages, appended template extension provides the following benefits:

- Templates are backward compatible
- Template ID changes are less frequent
- Expedited development time

In the event of an appended template extension, the following conditions will apply:

- Tags added to the **body** of the message will be added to the end of the body, before the repeating groups. **In this scenario, the TemplateID will not change, but the schema version will change.**
- The BlockLength field in the Binary Header or Group will increase in value to include the length of the new tag.
- Data fields can also be appended at the end of the repeating groups without updating the TemplateID, although the contents of the template and the schema version number will change. Each repeating group will include:
  - the new length of each entry in the repeating group
  - the number of entries in the repeating group

### Example: Template Before Appended Extension

For the example below, the schema version is 1.

```
<ns2:message id="99" name="ExampleTemplate99" description="ExampleTemplate" blockLength="9" semanticType="X" >
  <field name="Field1" id="9980" type="uint64" description="ExampleField1" offset="0" semanticType="int" />
  <field name="Field2" id="9981" type="Field2Type" description="ExampleField2" offset="8" semanticType="char" />
  <group name="NoMDEnties" id="268" description="Number of entries in Market Data message" blockLength="5" dimensionType="groupSize"
  >
    <field name="GroupField1" id="9990" type="Int32" description="ExampleGroupField1" offset="0" semanticType="int" />
    <field name="GroupField2" id="9991" type="GroupField2Type" description="ExampleGroupField2" offset="4" semanticType="char" />
  </group>
</ns2:message>
```

### Example: Template After Appended Extension

In the example below "ExampleField3" has been appended to the template 99 header. Additionally, the following updates were made to the schema:

- The schema version increases from 1 to 2.
- The header blockLength increases from 9 to 17. Eight bytes are added to the blockLength due to Field3 having a type uint64.

When client systems implement template extension in a decoder, the new field can be ignored and the location of the group can be determined within the message by leveraging the updated binary message header blockLength.

```
<ns2:message id="99" name="ExampleTemplate99" description="ExampleTemplate" blockLength="17" semanticType="X" >
  <field name="Field1" id="9980" type="uint64" description="ExampleField1" offset="0" semanticType="int" />
  <field name="Field2" id="9981" type="Field2Type" description="ExampleField2" offset="8" semanticType="char" />
  <field name="Field3" description="ExampleField3" id="9982" semanticType="int" offset="9" sinceVersion="2" type="uint64"/>
  <group name="NoMDEnties" id="268" description="Number of entries in Market Data message" blockLength="5" dimensionType="groupSize"
  >
    <field name="GroupField1" id="9990" type="Int32" description="ExampleGroupField1" offset="0" semanticType="int" />
    <field name="GroupField2" id="9991" type="GroupField2Type" description="ExampleGroupField2" offset="4" semanticType="char" />
  </group>
```

</ns2:message>

## Non-Appended Extension

For a non-appended template extension, the template is **not** backward compatible and the new template must be implemented. Examples of a non-appended extension include:

- fields inserted or removed from the middle of a block (message root or repeating group entry)
- encoding changes to existing fields, i.e. different data type or different size integer
- repeating group removed or inserted in the middle of a message



For both appended and non-appended message changes, customers will be notified of the schema file version change to indicate new information in the schema file.

## Schema Format

The schema file is constructed of several sections including the following:

**Primitive Type Encodings** - Valid encoding specifications for each of the integer primitive types.

**Simple Encodings** - Definition of specific values.

**Composite Encodings for FIX Types** - Specifies the range of valid values for a decimal field.

**Enumerations** - Conveys a single choice of mutually exclusive valid values.

### Message Schema

- **Message Description** - FIX Message Type Name.
  - ID - The Schema ID used identify the Template ID.
  - Semantic Type - FIX Message Type value for the message
  - Name - Identifies the message being encoded.
- **Field Descriptions**
  - Field Description - Field name
  - ID - FIX tag.
  - Semantic Type - FIX semantic data type and encoding type.
  - Name - FIX tag name.
  - Block Length - The size of the root level or the repeating group of the message in bytes.
  - Since Version - Version that the tag was added.
  - Offset - The number of bytes from the start of the message body or group to the first byte of the field.
  - Type - FIX semantic data type and encoding type.
- **Repeating Group Definitions**
  - Group Name - Unique identifier for a repeating group.
  - Field Description - Field Name.
  - ID - FIX Tag.
  - Semantic Type - FIX semantic data type and encoding type.
  - Name - FIX tag name.
  - Block Length - The size of the root level or the repeating group of the message in bytes.
  - Since Version - Version that the tag was added.
  - Offset - The number of bytes from the start of the message body or group to the first byte of the field.
  - Type - FIX semantic data type and encoding type.



### Repeating Group Number and Size

All repeating groups will be defined in the Market Data Schema File to facilitate distribution and usage of the file. The number and length of the repeating group will be defined in the top level repeating group tag; for example, tag 268-NoMDEntries, will indicate the number of repeating groups that follow and the common length of each.

### SBE Schema Header

The following is the layout of the SBE Schema header:

```
<ns2:messageSchema xsi:schemaLocation="http://www.fixtradingcommunity.org/pg/file/fplpo/read/1196759/simple-binary-encoding-rc2xsd  
SimpleBinary-RC2.xsd" byteOrder="littleEndian"
```

```
description="20140806"
```

```
semanticVersion="FIX5SP2"
```

```
version="5"
```

```
id="1"
```

package="mktdata"

**Schema Location** – Location of the SBE FPL Simple Binary Encoding Specification.

**Byte Order** - The order of the bytes stored in memory determines the interpretation of the data word. (**Little-endian** stores the *least significant byte* in the *smallest address*.)

**Description** – Date the Schema file was released.

**Semantic Version** – FIX Protocol version.

**Version** – The Version number of the Schema File.

**ID** – CME Group Internal Identifier

**Package** – CME Group Internal System Identifier