

ICS 35.240.40
CCS A 11

JR

**INDUSTRY STANDARD OF THE
PEOPLE'S REPUBLIC OF CHINA**

JR/T 0066.3-2019

Replacing JR/T 0066-2011

**Interbank market information exchange protocol - Part 3:
Presentation streaming layer
(English Translation)**

Issued on: January 8, 2019

Implemented on: January 8, 2019

Issued by: People's Bank of China

Table of Contents

Foreword
1 Scope
2 Terms and definitions
3 Syntax and Structure
4 Codec syntax
Appendix A (Informative) Summary of Error Codes

Foreword

JR/T 0066 "Interbank Market Business Data Exchange Agreement" is divided into 3 parts.

-Part 1. Grammar, structure and conversation layers;

-Part 2. Application layer;

-Part 3. The appropriate flow presentation layer.

This part is the third part of JR/T 0066.

This section is drafted in accordance with the rules given in GB/T 1.1-2009.

This section is proposed by China Foreign Exchange Trading Center and National Interbank Funding Center.

This section is under the jurisdiction of the National Financial Standardization Technical Committee (SAC/TC 180).

This section is responsible for drafting units. China Foreign Exchange Trading Center and National Interbank Funding Center.

Participated in the drafting of this section. Department of Science and Technology of the People's Bank of China.

1 Scope

This part of JR/T 0066 stipulates that the interbank market participants conduct data based on the interaction data of interbank market members at the session layer and the application layer.

Interbank Market Information Exchange Adapted for Streaming (IMAST).

This section applies to the front, middle and backstage of the foreign exchange market,

money market and derivatives market, and the front and middle stage of the bond market, excluding debt the background clearing and settlement part of the securities market.

2 Terms and definitions

The following terms and definitions apply to this document.

IMIX adapted for streaming (IMAST) A binary stream that compresses IMIX message data.

Encode

The process of converting IMIX messages into IMAST streams.

Decode

The process of converting an IMAST stream into a message.

Application type

The abstract concept of the IMIX session layer or application layer domain, component, or repeating group in the adaptive presentation layer.

Field instruction

The coding rules of the IMIX session layer or application layer domain, component, or repeating group in the appropriate flow presentation layer.

Template

A codec control structure implemented by a collection of several domain instructions in a scene.

Context

The operating environment for IMAST encoding or decoding.

3.1 IMAST protocol framework

The application mode of the flow-fit presentation layer is shown in Figure 1.

Figure 1 Application mode diagram of flow adaptation layer

The IMAST protocol framework relies on the template pointed to by the message to encode and decode to form an IMAST stream for network transmission.

A template is a codec control structure for an IMIX message.

Algorithms and binary code representation methods to control the encoding and decoding of binary streams corresponding to application messages. A template contains several domain pointers Order.

The domain directive specifies the encoding of domains, components, and repeating

groups in a message. Domain directives are grouped by name, application type, presence attribute, and operator to make.

The application type is the manifestation of the domain, component, or repeating group in the message at the appropriate flow presentation layer.

3.2 Overall structure of IMAST stream

The overall structure of the IMAST stream is defined using Extended Backus Normal Form (EBNF) syntax.

stream .. = message * | block * block .. = BlockSize message .. = segment * .. = PresenceMap TemplateIdentifier? (field | segment) * field .. = integer | string | Decimal | ByteVector the overall structure of IMAST is shown in Figure 2.

IMIX message

35 = 8 \u003cSOH\u003e Stream-adaptive coding IMAST stream Streaming decoding IMIX message 35 = 8 \u003cSOH\u003e template Domain directive Application Type template Domain directive Application Type

Figure 2 Overall structure of IMAST stream

The overall structure of the IMAST stream is described as follows.

- a) stream. An IMAST stream is a sequence of messages, or a sequence of blocks;
- b) Block. A block is a sequence containing a single or multiple messages. A block has a leading and is used to represent the block contains the block size in bytes of the message;
- c) Message. each message is represented by a segment (ie, a message segment);
- d) Segment. A segment consists of a header and a specific field.

An example of an IMAST stream is as follows.

An example of a market data request IMIX message is shown in Figure 3.

Figure 3 Example of market data request IMIX message

After encoding, the market data request IMAST stream is formed, where PAMP represents a placeholder map and TID represents a template identifier, as shown in Figure 4.

Message

Message

Segment

Message

Message

Message

Message

BlockSizeMessage

Message

Message

Message

Message

Message

Message

Placeholder map

PresenceMap

Template identifier

TemplatelIdentifier (optional) Field Field Subsection Segment Subsection Segment head
 BlockSizeMessage BlockSizeMessage 8 = IMIX.2.0 \u003cSOH\u003e 9 = 41
 \u003cSOH\u003e 34 = 2 \u003cSOH\u003e 35 = V \u003cSOH\u003e 49 =
 CFETS-RMB-CSTP \u003cSOH\u003e 56 = IH \u003cSOH\u003e 52 = .20151208-12. 00.
 00 \u003cSOH\u003e 263 = 2 \u003cSOH\u003e 1 46 = 1 \u003cSOH\u003e 48 = CNY
 \u003cSOH\u003e 10 = 167 Stream

Figure 4 Example of market data request IMAST flow

3.3 Byte Stop Bit

The IMAST stream is encoded using stop bit encoding rules, that is, each byte in the message has a stop bit. A stop bit encoding the body is a sequence of bytes, where the most significant bit of each byte indicates whether the next byte is real Part of the body.

An example of stop bits is shown in Figure 5.

Note: When the byte is the last byte of the stop bit entity, the stop bit is set to 1.

Figure 5 Example of stop bit

4.1 Overview

The encoding and decoding process depends on the template to which the message points. When encoding, the encoder advances the domains in the original message through the attributes of the domains in the template.

The encoding operation is performed, and a placeholder image is generated according to