



FIX Adapted for Streaming SM (FAST Protocol SM)

Session Control Protocol

Version 1.00

2006-05-10

Status of this Document

This document was revised as a result of a Market Data Optimization Working Group (MDOWG) review held on May 10, 2006 and several subsequent reviews.

This document is the official specification of the standard session control protocol for the FIX Adapted for STreaming(sm) protocol suite, commonly referred to as the FAST Protocol(sm). This protocol specified herein, is formally referred to as the **FAST Session Control Protocol (SCP) Version 1.0 (SCP 1.0)**.

Distribution

Distribution of this document is unlimited.

Copyright Notice

Copyright ©FIX Protocol Ltd. (2005, 2006)

Disclaimer

THE INFORMATION CONTAINED HEREIN AND THE FINANCIAL INFORMATION EXCHANGE PROTOCOL (COLLECTIVELY THE “FIX PROTOCOL”) ARE PROVIDED “AS IS” AND NO PERSON OR ENTITY ASSOCIATED WITH THE FIX PROTOCOL MAKES ANY REPRESENTATION OR WARRANTY, EXPRESS OR IMPLIED, AS TO THE FIX PROTOCOL (OR THE RESULTS TO BE OBTAINED BY THE USE THEREOF) OR ANY OTHER MATTER AND EACH SUCH PERSON AND ENTITY SPECIFICALLY DISCLAIMS ANY WARRANTY OF ORIGINALITY, ACCURACY, COMPLETENESS, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. SUCH PERSONS AND ENTITIES DO NOT WARRANT THAT THE FIX PROTOCOL WILL CONFORM TO ANY DESCRIPTION THEREOF OR BE FREE OF ERRORS. THE ENTIRE RISK OF ANY USE OF THE FIX PROTOCOL IS ASSUMED BY THE USER.

NO PERSON OR ENTITY ASSOCIATED WITH THE FIX PROTOCOL SHALL HAVE ANY LIABILITY FOR DAMAGES OF ANY KIND ARISING IN ANY MANNER OUT OF OR IN CONNECTION WITH ANY USER’S USE OF (OR ANY INABILITY TO USE) THE FIX PROTOCOL, WHETHER DIRECT, INDIRECT, INCIDENTAL, SPECIAL OR CONSEQUENTIAL (INCLUDING, WITHOUT LIMITATION, LOSS OF DATA, LOSS OF USE, CLAIMS OF THIRD PARTIES OR LOST PROFITS OR REVENUES OR OTHER ECONOMIC LOSS), WHETHER IN TORT (INCLUDING NEGLIGENCE AND STRICT LIABILITY), CONTRACT OR OTHERWISE, WHETHER OR NOT ANY SUCH PERSON OR ENTITY HAS BEEN ADVISED OF, OR OTHERWISE MIGHT HAVE ANTICIPATED THE POSSIBILITY, OF SUCH DAMAGES.

No proprietary or ownership interest of any kind is granted with respect to the FIX Protocol (or any rights therein) except as expressly set out in FPL’s copyright and acceptable use policy.

Acknowledgements

This specification is the result of the work of many people. Rolf Andersson did the initial drafting of the specification. Anders Furuheid, Bernt Sandén, Dave Rhodes, David Rosenborg, Greg Orsini, Göran Forsström, Jan Hård af Segerstad, Jonas Lundberg, Jonas Nordin, Matt Simpson, and Scott Atwell have

reviewed successive drafts of the specification. Subsequent review and feedback was provided by Jim Northey, Karl Botts, and Mark Weindling .

Abstract

This document is the official specification of the standard session control protocol for the FIX Adapted for STraming(sm) protocol suite (commonly referred to as the FAST Protocol(sm)). This protocol specified herein, is referred to as the FAST Session Control Protocol (SCP) Version 1.0 (SCP 1.0). SCP 1.0 provides a set of predefined messages that are used to initiate and control, via the exchange of state information, the exchange of FAST encoded messages. SCP 1.0 is an integral component of the FAST Protocol.

Table of Contents

| | | |
|-------|---|----|
| 1 | Introduction | 5 |
| 1.1 | Standard Naming and Abbreviations | 6 |
| 1.2 | Prerequisites | 6 |
| 1.3 | Terms and Acronyms | 6 |
| 2 | FAST Session Control Protocol Messages | 7 |
| 2.1 | Session Control Messages | 7 |
| 2.1.1 | FAST SCP 1.0 Messages | 7 |
| 2.1.2 | FAST SCP 1.0 Message Static Structure Diagram | 8 |
| 2.1.3 | TID Reserved Values..... | 8 |
| 2.2 | Message Definitions..... | 9 |
| 2.2.1 | FastHello1..... | 9 |
| 2.2.2 | FastAlert..... | 9 |
| 2.2.3 | FastReset..... | 10 |
| 2.3 | Example Message Flow - Successful Exchange of FastHello1 Messages | 11 |
| 2.4 | Failed Exchange of Hello Messages..... | 11 |
| 2.5 | Use of the FastReset message..... | 12 |

Document History

| Version | Date | Author | Description |
|---------|------------|----------------|---|
| 1.03 | 2006-05-05 | MDOWG | Incorporated reviewer feedback – including introduction of Block as a concept. |
| 1.02 | 2006-05-01 | MDOWG | Incorporated significant feedback from reviewers. |
| 1.01 | 2006-04-22 | MDOWG | Divided document into two main sections – background and the specification. |
| 1.00 | 2006-04-19 | MDOWG | Added Impetus, Applicability, Compliance, and Experimental Extensions sections to the document. Changed range of reserved values in the single byte TID affecting tables in sections 5.2.1 and 5.2.3 and corrected / changed examples accordingly., Converted messages to TD datatypes. Converted enum types to uInt32. |
| 0.98 | 2006-04-17 | Jim Northey | Corrected errors pointed out by Rolf, corrected the Compact Notation definition, including an example of Compact Notation. |
| 0.97 | 2006-04-17 | Rolf Andersson | Minor typo corrections |

| Version | Date | Author | Description |
|---------|------------|----------------|--|
| 0.96 | 2006-04-17 | Jim Northey | Compact Notation definition correction, FastHello1 naming inconsistencies, expanded the discussion on sessions to include late joining and framing. Added a summary table of session control messages, included Rolf's clarifications into the actual text from his comments, added glossary entries for terms Literal Value, Late Joining. Added an empty message table for FastReset for consistency and clarity, tied the FastAlert message definition to Appendix 1 for clarity. |
| 0.95 | 2006-04-17 | Rolf Andersson | Changed formatting of table in section 2. Corrected User's Guide document reference. Accepted all 0.94 changes. Highlighted (red color) a few remaining comments. |
| 0.94 | 2006-04-17 | Rolf Andersson | More minor corrections |
| 0.93 | 2006-04-16 | Jim Northey | Updated following review by Rolf Andersson |
| 0.92 | 2006-04-16 | Jim Northey | Modified number ranges for FAST Predefined Messages, revised discussion on version identification, renamed FastHello to FastHello1. |
| 0.91 | 2006-04-14 | Jim Northey | Basic copy editing and cleanup |
| 0.9 | 2006-04-03 | Matt Simpson | Revised and distributed for internal review |
| 0.8 | 2006-04-03 | Rolf Andersson | Clarified examples, removed experimental messages. |
| 0.7 | 2006-01-25 | Rolf Andersson | Updated following review by mdowg group. Capability negotiation removed, reset message added. Negotiate renamed to Hello. Notify renamed to Alert. Template exchange marked as experimental |
| 0.6 | 2006-01-03 | Rolf Andersson | More minor corrections |
| 0.5 | 2006-01-02 | Rolf Andersson | Updated following reviews |
| 0.4 | 2005-12-14 | Rolf Andersson | Updated following review by Scott Atwell |
| 0.3 | 2005-12-13 | Rolf Andersson | Document restructured and comments added following mdowg discussion. |
| 0.2 | 2005-12-11 | Rolf Andersson | Updates following Pantor internal review |
| 0.1 | 2005-12-05 | Rolf Andersson | First draft |

1 Introduction

SCP 1.0 provides a set of predefined messages that are used to initiate and control, the exchange of FAST encoded messages. SCP 1.0 is an integral component of the FAST Protocol Version 1.0. The use of SCP 1.0 for the transmission of FAST 1.0 Encoded Messages is optional. However, if an application requires session control, then SCP 1.0 is the appropriate and only FAST 1.0 compliant session control protocol that should be used.

Creation of the Session Control Protocol arose to address a need for session control to support FIX session encapsulation over FAST. This is commonly referred to as FAST tunneling. A later usage was identified to use the SCP FastReset message to explicitly identify Frame boundaries to communicate to the recipient of Fast Messages when the encoding state should be reset.

1.1 Standard Naming and Abbreviations

The following names and abbreviations should be used when referencing this standard in other documents included other standards specification documents.

Note that when discussing the protocol, you are strongly encouraged to include the version number of the standard so as to provide the context and eliminate future discrepancies as the version of the standards specification changes. For that reason, the version number is considered an integral part of the name and abbreviations of the standard.

| Standard naming and abbreviations | Usage |
|--|---|
| FAST Session Control Protocol (SCP) Version 1.0 | Full specification name |
| FAST SCP 1.0 | Full abbreviation - recommended for documents that may specify multiple standards or in which the context of the standard abbreviation may be ambiguous |
| SCP 1.0 | Contextual abbreviation for use within FAST Protocol specifications and documents that solely reference FAST and documents in which the context of the abbreviation is clear. |

1.2 Prerequisites

A basic understanding of computer network protocol theory and application is assumed. In order to fully benefit from this specification, the reader should be familiar with the FAST Protocol, as defined by the following documents.

FIX Adapted for Streaming (FAST Protocol) Field Encoding Specification, Version 1.0, 2006-01-11

FIX Adapted for Streaming (FAST Protocol) Transfer Encoding Specification, Version 1.0, 2006-01-13

Readers may also benefit from the following document, which is not a normative part of the specification, but provides a useful introduction to FAST.

A Basic User's Guide to Implementing FAST, Version 1.0, January, 2006

1.3 Terms and Acronyms

| Term / Acronym | Description |
|------------------|---|
| Acceptor | Computer application that accepts a FAST Session from an initiator. |
| Compact Notation | <p>A notation that represents the layout and format of a message as a series of fields delimited with the “ ” character and repeating groups with the “<” and “>” characters. Each field is displayed as printable ASCII string values representing strings, characters, integers, and decimals. Compact Notation is used primarily for documentation purposes.</p> <p>The following is an example of Compact Notation representing a FastAlert message. This example FastAlert message indicates there was a fatal protocol error.</p> <p style="text-align: center;">16001 14 1 Invalid message received</p> |
| FAST Tunneling | The use of a FAST session over which a standard FIX session is encoded to improve performance by reducing latency. |
| FE | Abbreviation for the FAST Field Encoding Specification Version 1.0 |

| Term / Acronym | Description |
|--|---|
| FAST Message Block (aka Block) | A sequence of one or more complete FAST Messages that are transmitted at one time. |
| FAST Message Block Size or Block Size (aka Block Size) | A Fast Unsigned Integer field specifying the length of a FAST Message Block (excluding the length of the FAST Message Block Size field itself) that can optionally be sent at the start of a FAST Message Block. |
| FAST Encoding Frame (aka Frame) | A sequence of one or more complete FAST MessageBlocks to which field encoding is applicable |
| Initiator | Computer application that initiates a FAST Session with an Acceptor. |
| Late Joining | In a scenario where there is one server and multiple clients receiving identical streams of FAST Messages in a FAST Session, clients need to be able to join mid-session (or rejoin). This ability to late join in the middle of a session is facilitated by dividing up the FAST Session into FAST Frames. Since by definition a FAST Frame contains the literal values for all encoded fields, clients can join the stream at any Frame boundary. |
| Literal Values | FAST field encoding relies on the client storing in a dictionary literal values which become the previous values for subsequent messages. FAST requires that the literal values for all field encoded fields be transmitted in the first FAST Message at the start of a FAST Frame. |
| PMAP | Presence map (refer to Transfer Encoding Specification Version 1.0 for details) |
| Previous Value | The previous value of a FAST Field. The previous values are used to determine the corresponding field value for the current message being decoded. |
| FAST Session | A bidirectional initiation, protocol negotiation, and subsequent exchange of FAST Messages |
| TE | Abbreviation for the Transfer Encoding Specification Version 1.0 |
| TID | Template ID (session local alias for a template name) Data type FAST Unsigned Integer |

2 FAST Session Control Protocol Messages

SCP 1.0 specifies the initiation of a FAST Session (FastHello1), the ability to indicate the start of a new *FAST Encoding Frame* (FastReset), the general notification of events that need to be communicated with counterparty (FastAlert) and the orderly termination of a session (FastAlert). There is no protocol capability negotiation functionality provided in SCP 1.0¹.

2.1 Session Control Messages

SCP 1.0 is composed of FAST Session Control Messages, whose templates are predefined and are identified using values from the *TID Reserved values*. The TID Reserved Values are divided into two ranges – a low value range for frequently used messages, referred to as High Frequency FAST Predefined Messages and a higher value range for messages that are used sparingly during a FAST session, referred to as Low Frequency FAST Predefined Messages.

2.1.1 FAST SCP 1.0 Messages

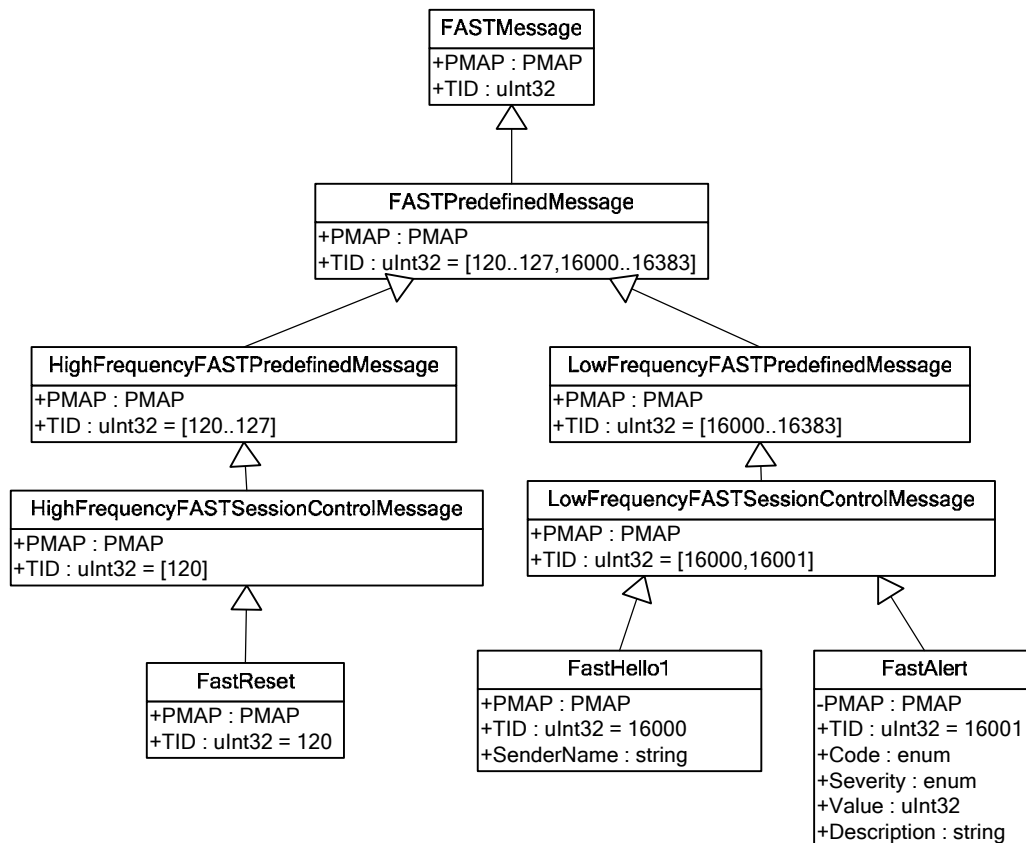
| Name | TID | Description |
|-----------|-----|-----------------------------------|
| FastReset | 120 | FAST Encoding Frame Reset message |

¹ Future versions of the FAST Session Control Protocol may address capability negotiation and dynamic exchange of templates between Initiator and Acceptor.

| Name | TID | Description |
|------------|-------|-----------------------------------|
| FastHello1 | 16000 | Hello Message for SCP Version 1.0 |
| FastAlert | 16001 | Alert message |

2.1.2 FAST SCP 1.0 Message Static Structure Diagram

The following UML analysis diagram is intended to communicate the conceptual structure of the FAST Session Control Protocol Messages.² This diagram explicitly defines FAST Session Control Messages as being FAST Predefined Messages, which in turn are FAST Messages.



2.1.3 TID Reserved Values

| Name | TID | Description |
|------------|-------------|--|
| - | 0 | (Reserved) |
| | 120-127 | High frequency FAST predefined messages |
| FastReset | 120 | FAST Encoding Frame Reset message |
| -- | 121-127 | Reserved for high frequency FAST predefined messages |
| | 16000 | Start of low frequency FAST predefined messages |
| FastHello1 | 16000 | Hello Message for SCP 1.0 |
| FastAlert | 16001 | Alert message |
| -- | 16002-16383 | Reserved for future low frequency FAST predefined messages |

² This UML diagram is provided as a conceptual model. The reader will likely concur that there is little value and maybe even some negative connotations in taking this UML diagram as is and using it for implementation. This is not an implementation diagram.

2.2 Message Definitions

All SCP 1.0 Messages are encoded according to the FAST 1.x specification.

2.2.1 FastHello1

FastHello1 is used at session initiation to exchange peer names. FastHello1 messages can be sent in parallel. It is recommended that the acceptor should wait for the initiator hello message before responding.

2.2.1.1 FastHello1 Message Body Layout

| Field | Type | Description |
|------------|--------|---|
| SenderName | string | Message originator name (Analogous to the FIX SenderCompID) |

2.2.1.2 SenderName

The SenderNames are determined by counter party agreement. There is no global name space or naming authority for SenderNames. Depending on the context, SenderName may or may not be unique.

2.2.1.3 Version Identification

The TID of the FastHello1 message used to initiate the FAST Session implicitly defines that SCP 1.0 is in use. The notion of version identification and capability negotiation are considered outside the scope of SCP 1.0 and will be addressed as a future release of the FAST Protocol.³

2.2.1.4 Compact Notation

16000 | AnyNameYouChoose |

2.2.1.5 Detailed Description

| Field | Value | Comment |
|------------|------------------|--|
| | | Message header |
| PMAP | | |
| TID | 16000 | FastHello1 |
| | | Message body |
| SenderName | AnyNameYouChoose | The Name of the Sender (originator) of the message |

2.2.2 FastAlert

FastAlert is used to send notifications to the peer. It is recommended that it is used before terminating a session to indicate the cause of termination. This practice will allow the communicating parties to identify the reason for disconnecting a session. Refer to Appendix 1 for a list of the enumerations for the Alert Severity and Alert Code enumerated fields.

2.2.2.1 FastAlert Message Body Layout

| Field | Type | Description |
|----------|--------|---|
| Severity | uInt32 | Alert Severity (see Appendix 1 - AlertSeverity) |
| Code | uInt32 | Alert Code (see Appendix 1 - AlertCode) |
| Value | uInt32 | Alert Value (depends on AlertCode) |

³ In terms of a general direction, current sentiment is to try and avoid explicit version numbers and instead take some approach to identify individual capabilities.

| | | |
|-------------|--------|-------------------|
| Description | string | Alert Description |
|-------------|--------|-------------------|

2.2.2.2 Compact Notation

16001|1|13|Ø|Not Authorized|

2.2.2.3 Detailed Description

| Field | Value | Description |
|-------------|------------------|------------------------|
| | | Message header |
| PresenceMap | | |
| MessageTID | 16001 | FastAlert |
| | | Message body |
| Severity | 1 | Fatal |
| Code | 13 | Not Authorized |
| Value | NULL | |
| Description | "Not Authorized" | Descriptive error text |

2.2.3 FastReset

FastReset is used to communicate the start of a new of new FAST Encoding Frame. The FastReset message contains no data fields. The field encoding history is set implicitly at an initial (reset) state when a new transport (e.g. TCP) session is established. It is therefore not necessary to send a reset message at session establishment time.

2.2.3.1 FastReset Message Body Layout

| Field | Type | Description |
|-------|------|---|
| | | <i>The FastReset contains no fields</i> |

2.2.3.2 Compact Notation

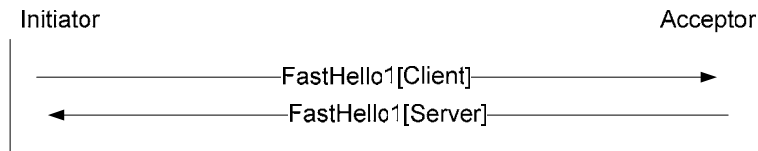
120|

2.2.3.3 Detailed Description

| Field | Value | Comment |
|-------|-------|-----------------------|
| | | Message header |
| PMAP | | |
| TID | 120 | FastReset |
| | | Message body |

2.3 Example Message Flow - Successful Exchange of FastHello1 Messages

Message Interactions



2.3.1.1 Initiator FastHello1 Message

Compact Notation

16000|CLIENT|

Detailed Description

| Field | Value | Description |
|------------|--------|----------------|
| | | Message Header |
| PMAP | | |
| TID | 16000 | FastHello1 |
| | | Message Body |
| SenderName | CLIENT | |

2.3.1.2 Acceptor FastHello1 Message

Compact Notation

16000|SERVER|

Detailed Description

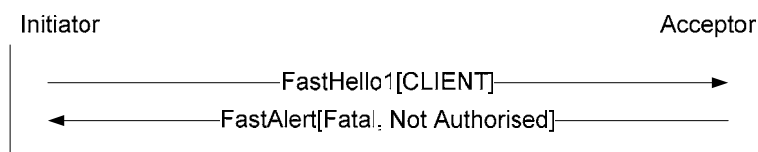
| Field | Value | Description |
|------------|--------|----------------|
| | | Message header |
| PMAP | | |
| TID | 16000 | FastHello1 |
| | | Message body |
| SenderName | SERVER | |

2.4 Failed Exchange of Hello Messages

Scenario

The Acceptor refuses to establish the connection (at the application level)

Message Interactions



2.4.1.1 Initiator FastHello1 Message

Compact Notation

16000|CLIENT|

Detailed Description

| Field | Value | Description |
|-------------|--------|-----------------------|
| | | Message header |
| PresenceMap | | |
| MessageTID | 16000 | FastHello1 |
| | | Message body |
| SenderName | CLIENT | |

2.4.1.2 Acceptor FastAlert Message

Compact Notation

16001|1|13|Ø|Not Authorized|

Detailed Description

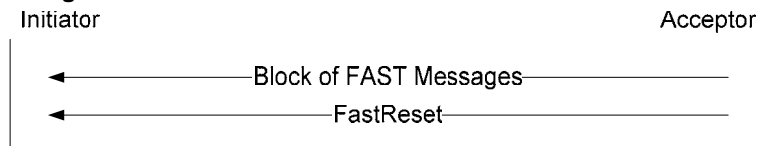
| Field | Value | Description |
|-------------|------------------|------------------------|
| | | Message header |
| PresenceMap | | |
| MessageTID | 16001 | FastAlert |
| | | Message body |
| Severity | 1 | Fatal |
| Code | 13 | Not Authorized |
| Value | NULL | |
| Description | "Not Authorized" | Descriptive error text |

2.5 Use of the FastReset message

Scenario

Acceptor transmits one or more FAST Message Blocks, followed by a FastReset message

Message Interactions



2.5.1.1 Acceptor sends a FastReset after sending a Block of one or more FAST Messages

Compact Notation

120

Detailed Description

| Field | Value | Description |
|-------------|-------|-----------------------|
| | | Message header |
| PresenceMap | | |
| MessageTID | 120 | FastHello1 |
| | | Message body |

Appendix 1 – Assigned Codes

Assigned codes are maintained by FPL's Global Technical Committee (FPL/GTC).

(Please refer to: <http://www.fixprotocol.org/fast>)

AlertCode

Alert codes are used in the FastAlert message to identify the type of notification.

| Code | Short Name | Description |
|------|------------|------------------------|
| 0 | | (Reserved) |
| | | Information |
| | | |
| | | |
| | | Errors |
| 11 | TNOTSUPP | Template not supported |
| 12 | TUNKNOWN | Template unknown |
| 13 | EAUTH | Unauthorized |
| 14 | EPROTO | Protocol Error |

AlertSeverity

Alert severities are used in FastAlert messages to identify the severity of each alert.

| Code | Short Name | Description |
|------|------------|------------------------------------|
| 0 | | (Reserved) |
| 1 | FATAL | Fatal (session must be terminated) |
| 2 | ERROR | Error (operation failed) |
| 3 | WARN | Warning (operation succeeded) |
| 4 | INFO | Information (operation succeeded) |