



FIX Global Technical Committee

FIX Technical Standard Lifecycle and Processes

December 20, 2022

[Revision #4]

Document Status: FINAL

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.



This work is licensed under a [Creative Commons Attribution-NonCommercial 4.0 International License](https://creativecommons.org/licenses/by-nc/4.0/).

No proprietary or ownership interest of any kind is granted with respect to the FIX Protocol (or any rights therein).

Copyright 2003-2023 FIX Protocol Limited, all rights reserved.

Table of Contents

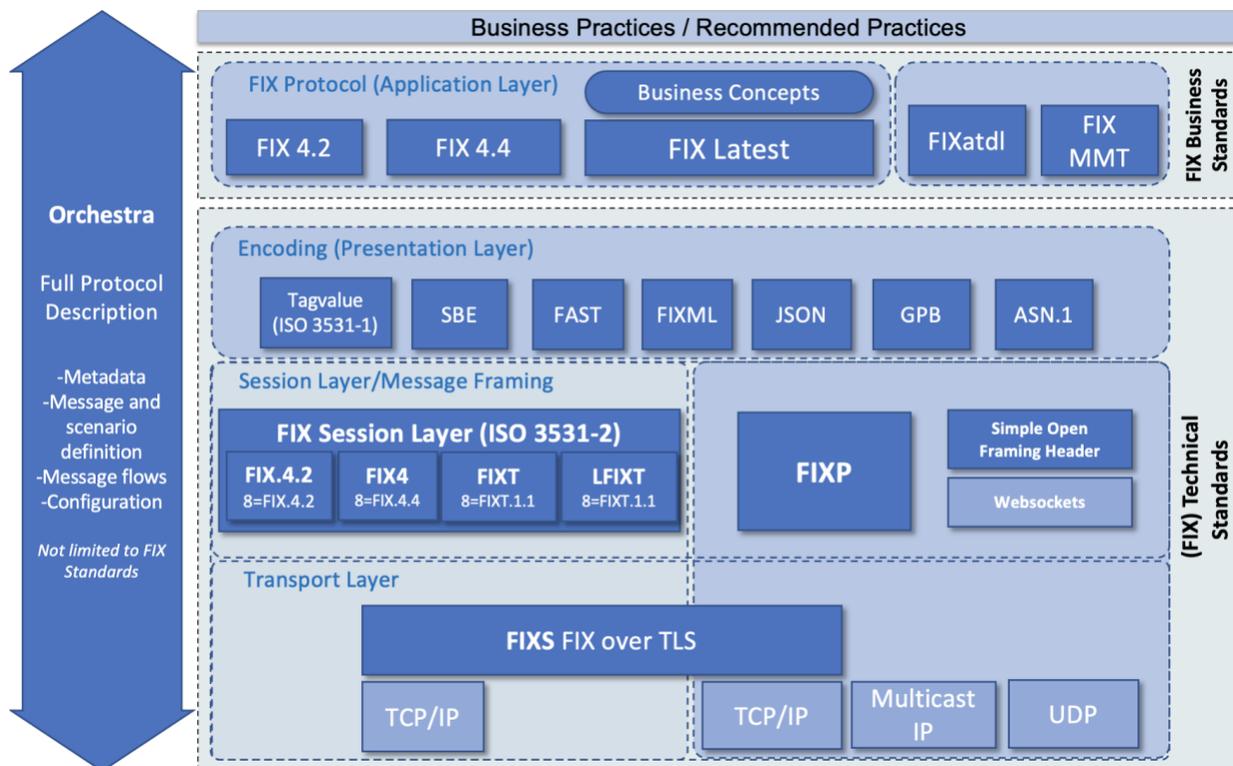
Document History	4
1 Introduction	5
2 Lifecycle	6
2.1 Version	6
2.2 Release Candidate (RC)	6
2.3 Beta Version (BETA)	6
2.4 Technical Standard (STANDARD)	7
2.5 Technical Standard Errata (ERRATA)	7
3 Processes	8
3.1 Development and Maintenance	8
3.1.1 Compilation of Documents	8
3.1.2 Release Candidate versus Beta Version	8
3.2 Submission to the GTC	8
3.3 Review by the GTC	9
3.4 Public Review	9
3.5 Approval and Publication	9
Appendix A – FIX Technical Standards	10

1 Introduction

The FIX Protocol organization provides a number of Technical Standards to the FIX Trading Community. These are subject to a well-defined process and lifecycle defined by the FIX Global Technical Committee (GTC) by this document.

The objective is to avoid inconsistencies that may be technical, documentation or level of quality in nature. A well-defined process ensures the maturity of the final standard and should always include actual implementations by more than one party.

FIX Technical Standards are defined for every layer of the FIX Technical Stack as shown in the following diagram. The only exception is the FIX Application Layer that is subject to its own process and lifecycle (Gap Analysis and Extension Pack process) as defined in a separate [document](#).



2 Lifecycle

A FIX Technical Standard has a name, a version number, and a document with its Technical Specification. Prior to becoming final, every version evolves over a number of revisions as described below. A Technical Standard is maintained by a FIX Working Group (WG), Subgroup, Committee or Subcommittee (the term “WG” is used in this document to represent any of them) that maintains it and decides when to take the next step in the evolution and to submit the specification to the FIX Global Technical Committee to start the formal approval process.

The Technical Specification documents for FIX Technical Standards are licensed under a [Creative Commons Attribution-NoDerivatives 4.0 International License](#).

2.1 Version

A FIX Technical Standard evolves over time by means of versions that enhance the scope and/or capabilities. FIX uses major and minor version numbers (Version <major>.<minor>). A new minor version number is chosen whenever changes are additive (optional) and hence non-breaking. A new major version may be used if the scope is significantly increased and must be used if backward compatibility cannot be maintained. The WG proposes a new version number as part of its submission to the GTC whereby it can be approved or rejected.

2.2 Release Candidate (RC)

The initial draft of the specification is the first Release Candidate (a.k.a. RC1). The WG considers it to be ready for implementation. The objective is to inform the wider FIX community and the general public that a new Technical Standard is being developed by FIX. Interested parties are able to review the details of the specification and engage in Proof-of-Concept (PoC) activities.

Additional Release Candidates (RC2, RC3,...) may follow the initial one in order to enhance the scope and capabilities of the standard. They may also serve to discard features introduced with previous Release Candidates, i.e. backward compatibility is not guaranteed between Release Candidates.

There is no limit to the number of Release Candidates for a given version of a Technical Standard. It is not recommended to use Release Candidates for the production version of an application due to the possibility of significant changes being made before becoming final.

2.3 Beta Version (BETA)

The Beta Version is almost identical to the last Release Candidate. The reason is that a Beta Version cannot be subject to changes that have a significant impact on backward compatibility. The transition to a Beta Version indicates that the WG believes it to be of suitable quality and fit for purposefulness to be adopted.

The community is encouraged to implement and test the specification in order to identify any issues such as ambiguities due to a lack of granularity in the document.

The Beta Version is published for a period of twelve months and then transitioned to a Technical Standard unless serious issues are raised from an implementation perspective. The GTC may then decide to send the specification back to the Release Candidate stage for the WG to make corrections.

2.4 Technical Standard (STANDARD)

This is the final specification of a given version of a Technical Standard. The scope is identical to the Beta Version, but the specification may contain additional or modified text, e.g. for clarification purposes. Existing implementations of the Beta Version should be analyzed to confirm their continued compliance with the final specification.

2.5 Technical Standard Errata (ERRATA)

The final Technical Standard (STANDARD) may still contain minor errors, e.g. typos or issues. Updates of a Technical Standard are possible by means of Errata releases. The version number of the Technical Standard does not change, it merely has the suffix “Errata”, e.g. “Version 1.0 Errata” and a date (not a sequence number).

There is no limit to the number of Errata releases for a given version of a Technical Standard. They do not apply to Release Candidates or a Beta Version.

3 Processes

3.1 *Development and Maintenance*

The FIX Protocol organization develops all Technical Standards as public repositories in [GitHub](#). The Technical Specification is written in markdown and treated like source code. It allows tracking of all changes made across the lifecycle of a Technical Standard. Standard GitHub processes such as issues and pull requests support the lifecycle activities. Members of the WG responsible for the Technical Standard are expected to use these processes for development and maintenance.

Registered GitHub users can enter issues for discussion by the WG and as a preliminary step to a pull request. The pull request then contains the actual change proposal and can be reviewed and approved by members of the WG. Reviewers may ask for changes before accepting a pull request. This process is public but does not replace the public review as described in Chapter 3.4. The scope of the latter is the complete specification and allows interested parties that are not active on GitHub to provide feedback.

The FIX Technical Support Team has the overall responsibility for the management of the GitHub repositories. This specifically relates to merging of pull requests to update the specification and the definition of releases and/or tags in GitHub.

3.1.1 **Compilation of Documents**

The Technical Specification in markdown is converted to a Microsoft Word document and an html document by means of open-source tools used as part of an automated script. The Microsoft Word document is generated on the basis of a standard style document to obtain an identical layout across all FIX Technical Standards. It is then converted to a PDF document with navigational bookmarks for general publication.

3.1.2 **Release Candidate versus Beta Version**

Development does not pause during periods of public review, i.e. changes for the next Release Candidate (or eventually Beta Version) can be made immediately after the previous one has been published for review. Introducing major changes between the last Release Candidate and the Beta Version carries the risk that these changes can only be reversed with the next version of the Technical Standard. This may need to be a major new version, e.g. V2.0 instead of V1.1, if the change breaks backward compatibility.

3.2 *Submission to the GTC*

The WG developing or maintaining a Technical Standard decides when to make a submission to the GTC via the respective GTC Subcommittee (see Appendix A). The GTC Subcommittee may conduct its own review meeting prior to the GTC submission but does not need to formally approve the submission.

Any submission to the GTC must include the Technical Specification and a Technical Proposal, a document based on a standard [template](#). It is only used for the review and approval process and is not considered to be part of the Technical Standard itself.

The Technical Proposal provides background information as well as a high-level description of the Technical Standard itself. It can give credit to the authors and provide links to related material. The main part of the Technical Proposal describes the changes that have been made since the last Release Candidate (or Beta Version).

3.3 Review by the GTC

The GTC schedules submissions for Technical Standards as part of its regular (monthly) meetings. It is also possible to schedule an ad-hoc GTC meeting to avoid delays in the process. A representative from the submitting WG is expected to present the Technical Proposal document during the meeting. The main focus is on the changes that have been made since the last Release Candidate (or Beta Version).

The Technical Specification is provided as part of the meeting material but not reviewed in detail. Interested parties may still raise issues pertaining to the Technical Specification during the meeting.

The participants of the GTC meeting decide whether to approve the submission for publication or not. Publication refers to all submitted artefacts and may be for public review (for Release Candidates) or for information (Beta Version and Technical Standard). Issues and concerns are captured during the meeting and may require minor changes to the Technical Proposal and/or Technical Specification prior to publication.

The GTC may reject the submission in its entirety and request significant changes from the submitting WG. Optionally, a WG may involve selected resources from the GTC ahead of the submission to avoid a rejection of the proposal.

3.4 Public Review

A public review period of 90 days is required for a Release Candidate approved during the GTC meeting where it was presented. There are two main options for the community to provide feedback:

- **FIX Discussion Forum:** FIX posts an entry to the [public forum](#) with links to the artefacts submitted to the GTC. Registered users (FIX membership is not required) can provide feedback as comments to the entry.
- **FIX GitHub Repository:** GitHub offers issues and pull requests to registered users (FIX membership is not required) to provide feedback.

Beta Version (BETA) and Technical Standards (STANDARD or ERRATA) are not subject to public review. They are provided for information on the FIX discussion forum. However, comments can always be provided in GitHub as issues to be applied to the next version of the Technical Standard. Minor errors such as typos may still be considered. This is also possible when the official review period of a Release Candidate has ended. Feedback is always applied to the next Release Candidate (RCn+1) and not to the one in review.

3.5 Approval and Publication

Only the final Technical Standard (STANDARD) requires the approval of the GTC Governance Board based on a simple majority of votes. Members of the GTC Governance Board are also members of the GTC and, if they are interested, can get involved much earlier in the process as part of that role. The public review is also not limited to people outside of the GTC, e.g. when a GTC member was unable to attend the GTC meeting where the submission was reviewed and approved for public review.

Release Candidates, Beta Versions and Technical Standards (STANDARD or ERRATA) are published on the FIX website (<https://www.fixtrading.org/>) for download (PDF document) and in an online format. They are both generated from the same source in the GitHub repository and should hence be identical. Differences may only occur in terms of layout. However, only the latest publication of the latest version of a Technical Standard (final or not) is published in an online format. Downloads are also available for previous versions and all Release Candidates, the Beta Version, and the Technical Standard.

Appendix A – FIX Technical Standards

The following table lists key information for all FIX Technical Standards.

Layer	Technical Standard	GTC Subcommittee	GitHub Repository (https://github.com/FIXTradingCommunity)
Meta-Data	Orchestra	Orchestra	fix-orchestra-spec
Application	FIXatdl	FIX Protocol	fixatdl-specification
Encoding	tagvalue	FIX Standards	fix-session-layer-standards
Encoding	FIXML	FIX Standards	FIXML
Encoding	SBE	FIX Standards	fix-simple-binary-encoding
Encoding	FAST	FIX Standards	FAST
Encoding	JSON	FIX Standards	fix-json-encoding-spec
Encoding	GPB	FIX Standards	fix-gpb-encoding-spec
Encoding	ASN.1	FIX Standards	(not available)
Session	FIX Session Layer	FIX Standards	fix-session-layer-standards
Session	FIXP	FIX Standards	fixp-specification
Session	SOFH	FIX Standards	fix-simple-open-framing-header
Session	FIXS	FIX Standards	fixs-specification