



**FINANCIAL INFORMATION
EXCHANGE PROTOCOL
(FIX)**

Version 5.0 Service Pack 2 - [Errata](#)

VOLUME 2 – TRANSPORT PROTOCOLS

~~April 2009~~ [August 18, 2011](#)

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 FIX Protocol Limited's Copyright and Acceptable Use Policy.

© Copyright 2003-~~2009~~[2011](#) FIX Protocol Limited, all rights reserved

REPRODUCTION

FIX Protocol Limited grants permission to print in hard copy form or reproduce the FIX Protocol specification in its entirety provided that the duplicated pages retain the "Copyright FIX Protocol Limited" statement at the bottom of the page.

Portions of the FIX Protocol specification may be extracted or cited in other documents (such as a document which describes one's implementation of the FIX Protocol) provided that one reference the origin of the FIX Protocol specification (<http://www.fixprotocol.org>) and that the specification itself is "Copyright FIX Protocol Limited".

FIX Protocol Limited claims no intellectual property over one's implementation (programming code) of an application which implements the behavior and details from the FIX Protocol specification.

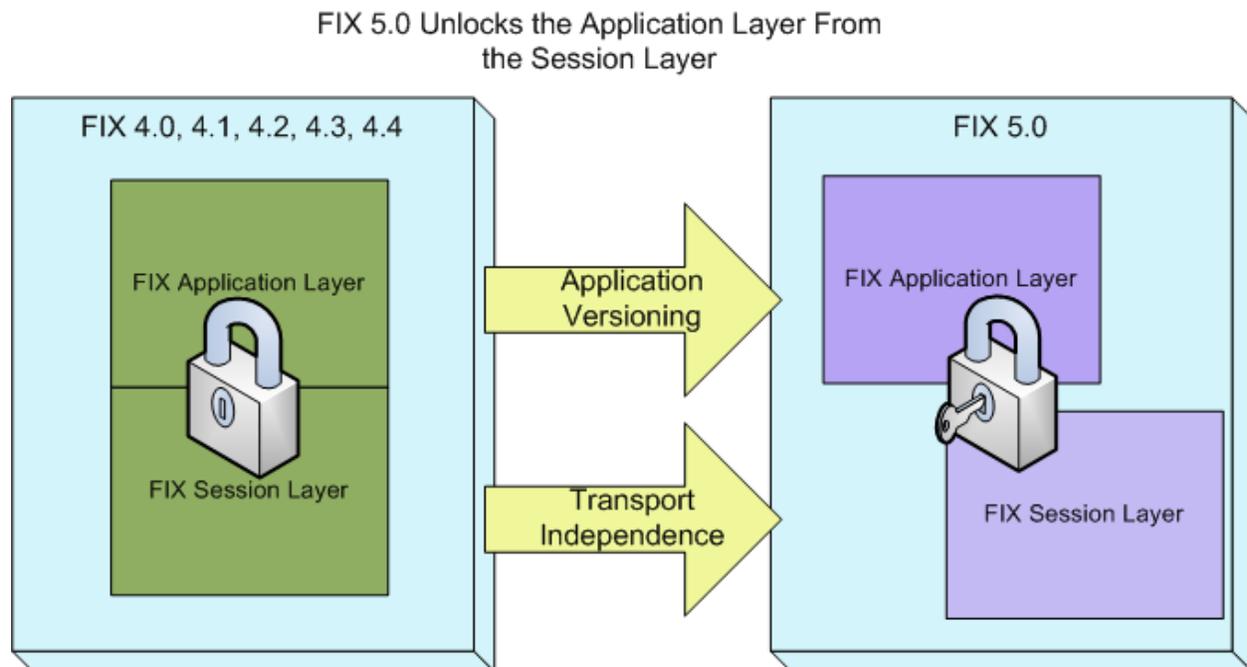
Contents – Volume 2

INTRODUCTION	21
TRANSPORT INDEPENDENCE (TI) FRAMEWORK.....	28
APPLICATION VERSIONING.....	33
SERVICE PACK MANAGEMENT	37
EXTENSION PACK MANAGEMENT	40
USE CASE 1 – FORMAL FIX 5.0 RELEASE.....	46
USE CASE 2 – FIX 5.0 RELEASE W/ SUPPORT FOR LEGACY	57
USE CASE 3 – TRANSPORT INDEPENDENCE	60
TRANSPORT PROTOCOLS.....	63
FIX SESSION PROTOCOL.....	64
FIX USING A MULTICAST TRANSPORT	65
FIX USING THE FAST SESSION CONTROL PROTOCOL (SCP).....	68
FIX USING MQSERIES.....	71
FIX USING WEB SERVICES.....	74

Introduction

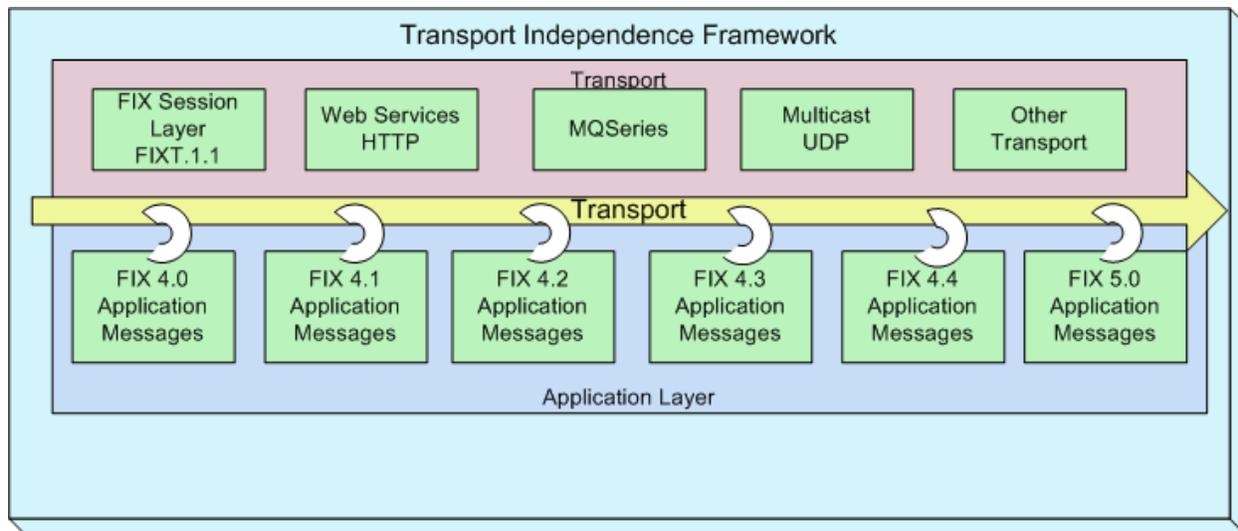
With the release of FIX 5.0 in December 2006, the FPL Global Technical Committee (GTC) introduced a new framework, the transport independence (TI) framework, which separated the FIX Session Protocol from the FIX Application Protocol. Under the TI framework the application protocol messages can be sent over any suitable session transport technology (e.g. WS-RX, MQ, publish/subscribe message bus), where the FIX Session Protocol is one of the available options as a session transport for FIX application messages. From this release forward the FIX Application layer and the FIX Session layer will have their own versioning moniker. The FIX Application layer will retain the traditional version moniker of "FIX x.y" while the FIX Session layer will utilize a new version moniker of "FIXT x.y" (note that the version numbers will be independent of each other). The diagram below illustrates how previously the FIX Session layer was tightly coupled to the Application layer. With the advent of Application Versioning and Transport Independence, the FIX Session and Application layers have been decoupled and are now independent.

This Volume of the FIX Protocol Specification will not discuss any particular session protocol, but will provide pointers to documents that describe the various transport protocols that the GTC has investigated and provided recommendation on how they should be used within the TI framework.



Transport Independence (TI) Framework

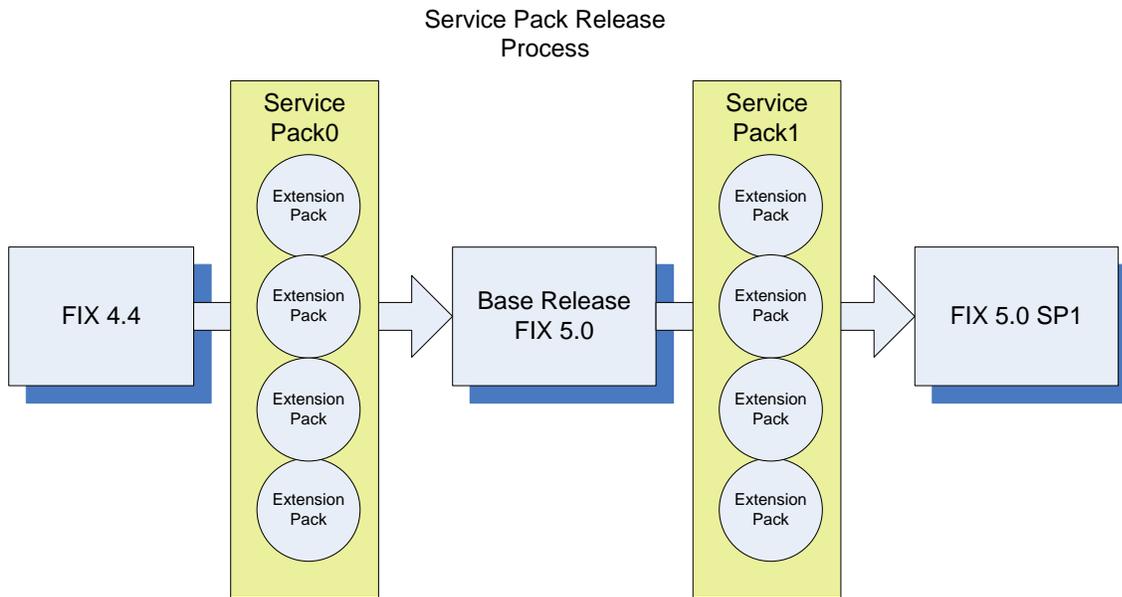
The transport independence (TI) framework separates the previously coupled FIX Session layer from the FIX Application layer. Under this framework the FIX Application Protocol can use any transport technology in addition to the FIX Session Protocol. The diagram below illustrates how various transport mechanisms, including the FIX Session layer, can be used to carry the full suite of FIX Application versions.



To support this framework a key new field has been added called *AppVerID* (application version ID, tag 1128). Depending on the use case *AppVerID* may be optional or required. Additionally, the FIX field *BeginString* will no longer identify the FIX application version, but identifies the FIX Session Protocol version. The sections below discuss the four main use cases supported by the TI framework.

Application Versioning

Application Versioning allows extensions to the current base application version to be applied using a formal release process. Extension Packs represent the individual gap analysis proposals submitted to the GTC for review and approval. Extension Packs are grouped into Service Packs and are applied to the base application version, usually the most current FIX application version. A new application version is formed when a new Service Pack is applied to a base version. In the diagram below, FIX 4.4 has been extended via Service Pack 0, forming a new application version called FIX 5.0. As new Extension Packs are approved they will be grouped into Service Pack 1 which is then released to form the next application version identified as FIX 5.0 SP1. These application versions are expressed using the new tag *AppVerID*.



Service Pack Management

ApplVerID is an enumerated field. These enumerations are used to express prior versions of FIX inclusive of FIX 4.0, 4.1, 4.2, 4.3 and 4.4 as well as the most recent version, FIX 5.0. Going forward, service packs will be applied to the base version, in this case FIX 5.0, and will be identified as FIX Version + Service Pack . This means that FIX 5.0 will be represented as an enumeration (7) rather than as an actual value in the ApplVerID field. Service Pack identifiers will consist of the base FIX version, the service pack number for that version, and the date the service pack was released. For example, the assigned value for service pack 1 may be “FIX 5.0 SP1 June 30, 2007”.

Extension Pack Management

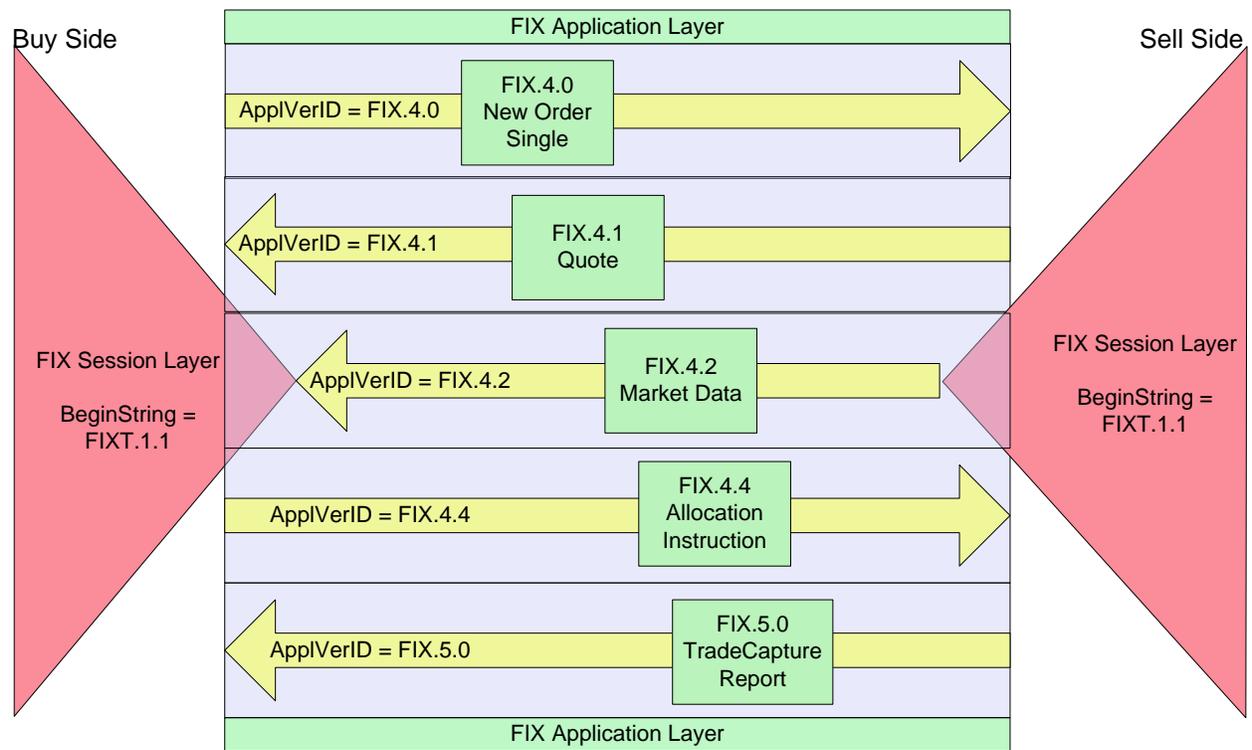
Extension Packs are the building blocks of a Service Pack and represent specific functional proposals that have been presented to the GTC. Prior to the release of a Service Pack, Extension Packs are applied to the most recent version of the repository so that they can be used at the point they become available. Extension Packs are applied to the repository in a cumulative manner and will at some point culminate in a Service Pack release. Extension Packs management will be conducted as follows:

1. Extension Packs will be assigned a unique, sequential number at the point they are approved by the GTC
2. Extension Packs are applied to the most recent version of the repository and may be inclusive of prior Extension Packs
3. At the point an Extension Pack has been applied, the updated repository, schema, and message tables will be available
4. When implementing a specific Extension Pack, the field CustomApplVerID (1129) will be used to specify the Extension Pack Identifier
5. User's of an Extension Pack need not implement other Extension Packs present in the repository. Rules of engagement need to be bilaterally agreed on.

Use Case 1 – Formal FIX 5.0 Release

This is the 'GTC approved' approach which separates the FIX session layer from the application layer, provides support for application versioning, and creates a platform for transport independence. This approach will treat the FIX session like 'any other' transport and allow the unambiguous use of any application version via the *AppVerID* field. A value of FIXT.1.1 in the *BeginString* of the FIX Session will indicate that application versioning is in effect and the version should be determined either through the Logon's *NoMsgType* repeating group or the *AppVerID* field. Future extensions to the session layer or application layer will be supported independent of each other as point releases to *BeginString* and *AppVerID*, respectively. Major Tags describing the session and application versions are: *BeginString*=FIXT.1.1 (or later versions) and *AppVerID*=7 (FIX.5.0¹) or later versions. A *BeginString*=FIX.5.0 (or later versions) will not be valid.

The diagram below illustrates how the new FIXT.1.1 Session layer makes use of the *AppVerID* in the Application layer in order to support a broad set of application versions.²



¹ The value FIX.5.0 is represented by enumeration value 7

² FIX.4.0, FIX.4.1, FIX.4.2, FIX.4.3, FIX.4.4, FIX.5.0, FIX.5.0.SP1, FIX.5.0.SP2 are represented using enumerations

Use Case 2 – FIX 5.0 Release w/ Support for Legacy

This approach builds on Use Case 1 above but makes a provision for legacy installations by allowing the application version ID of the legacy version to be specified as a default in the Logon message in the *DefaultAppVerID* (tag 1137) field. This would apply to those all messages not carrying the *AppVerID* field or overridden through the Logon's *NoMsgType* repeating group. Major Tags describing the session and application versions are: *BeginString*=FIXT.1.1 (or later versions) and *DefaultAppVerID* set to other earlier versions of FIX. It should be noted that *DefaultAppVerID* can also be used in Use Case 1 in a similar manner to default the session's application version to FIX 5.0 or later versions - this eliminates the need to have to specify *AppVerID* in every message if the session will primarily carry FIX 5.0 application messages.

Use Case 3 – Transport Independence

This is a standalone release of FIX 5.0. A transport other than the FIX session protocol is used and all application messages must carry the *AppVerID* field to identify the version of the FIX application message. *BeginString* is not used.

Under this use case, session level messages as defined in FIXT 1.1 are not available and are thus unsupported message types. Note that this includes the session level Reject message. In situations where rejection of messages may be needed, the alternate transport might offer a similar reject message that can be used to reject messages that violate session-level rules. If this is not the case, the Business Message Reject message should be used instead.

Transport Protocols

FIX Session Protocol

The FIX Session Protocol Version 1.1 specification can be found at <http://www.fixprotocol.org/specifications>). This is the traditional FIX Session Protocol with additions to support TI.

FIX Using a Multicast Transport

A best practices guide to using a multicast transport can be found at http://fixprotocol.org/documents/2519/FIX_Multicast_Transport_v1.0.pdf . Although this document specifically discusses the dissemination of FIX Market Data messages over a Multicast Transport, the concepts discussed can be use on any FIX messages. However, users are cautioned that using a Multicast Transport for trade transactions is not recommended.

FIX Using the FAST Session Control Protocol (SCP)

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. The complete SCP 1.0 specification can be found at http://fixprotocol.org/documents/2375/FAST_SessionControlProtocol_v1.00.pdf

FIX Using MQSeries

The FIA standards working group has drafted a set of proposed standards for the use of IBM WebSphere MQ (formerly MQSeries) to facilitate intercommunication between its members. The recommended standards will be based on best practices germane to the post-trade environment and the post-trade messages exchanged by participating counterparties.

Application Guidelines include recommendations for the use of the MQI standard WMQ application programming interface for sending and receiving messages. *Administrative Guidelines* include recommendations for defining the Queue Managers and the objectsthrough which intercommunication between participants is achieved. The document can be found at <http://fixprotocol.org/documents/1869/MOStandardizationrequirements5.doc>

FIX Using Web Services

Currently the Web Services Working Group is continuing its investigation of web services while the group waits for OASIS to approve the WS-RX specification. The Web Service Working Group is looking at WS-RX as the specification to base FIX's WS policies on. For current information on the Web Services Working Group please see the group's web page at: http://www.fixprotocol.org/working_groups/wswg . To participate in this working group please contact the FPL Program Office at fpl@fixprotocol.org