



Orchestra Standard - Technical Proposal

Version 1.1 RC2

November 12, 2024

V0.2

Proposal Status: Submitted

For Global Technical Committee Governance Internal Use Only

Submission Date	Nov 26, 2024	Control Number	
Submission Status	Submitted	Ratified Date	
Primary Contact Person	Hanno Klein FIX Technical Director	Release Identifier	

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.

DRAFT OR NOT RATIFIED PROPOSALS (REFER TO PROPOSAL STATUS AND/OR SUBMISSION STATUS ON COVER PAGE) ARE PROVIDED "AS IS" TO INTERESTED PARTIES FOR DISCUSSION ONLY. PARTIES THAT CHOOSE TO IMPLEMENT THIS DRAFT PROPOSAL DO SO AT THEIR OWN RISK. IT IS A DRAFT DOCUMENT AND MAY BE UPDATED, REPLACED, OR MADE OBSOLETE BY OTHER DOCUMENTS AT ANY TIME. THE FIX GLOBAL TECHNICAL COMMITTEE WILL NOT ALLOW EARLY IMPLEMENTATION TO CONSTRAIN ITS ABILITY TO MAKE CHANGES TO THIS SPECIFICATION PRIOR TO FINAL RELEASE. IT IS INAPPROPRIATE TO USE FIX WORKING DRAFTS AS REFERENCE MATERIAL OR TO CITE THEM AS OTHER THAN "WORKS IN PROGRESS". THE FIX GLOBAL TECHNICAL COMMITTEE WILL ISSUE, UPON COMPLETION OF REVIEW AND RATIFICATION, AN OFFICIAL STATUS ("APPROVED") OF/FOR THE PROPOSAL AND A RELEASE NUMBER.

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

Copyright 2003-2024 FIX Protocol Limited, all rights reserved.

Table of Contents

Document History	4
1 Introduction	5
1.1 Roadmap	5
1.2 Authors	5
2 Requirements	6
2.1 Business Requirements	6
2.1.1 Scenario Relationships (#A-228)	6
2.1.2 Elaborate on message mapping to scenarios (#B-52)	7
2.1.3 Update recommendation for Dublin Core terms (#B-46)	7
2.1.4 Add and define documentation purposes (#A-229)	8
2.1.5 Increase maximum length of MsgType attribute (A-#208)	8
2.2 Technical Requirements for Repository Schema	9
2.2.1 Correct reference to mapped datatype element (B-#43)	9
2.2.2 Allow names in correlation and assignment references (A-#187)	9
2.2.3 Remove attribute applVerID (A-#186)	9
2.2.4 Remove unused attribute group for messages (A-#189)	9
2.2.5 Remove unused simple types (#A-195)	10
2.2.6 Move datatype definition to repositoryType.xsd (#A-190)	10
2.2.7 Correct cardinality restriction of groups (#A-203)	11
2.2.8 Define attribute changeType (#B-42)	11
2.2.9 Make selected section and category attributes required (#A-191)	12
2.2.10 Mark section of datatype mapping as non-normative (#B-47)	12
2.3 Technical Requirements for Interfaces Schema	13
2.4 Open issues	13
3 Issues and Discussion Points	14
4 References	14
5 Relevant and Related Standards	14
6 Intellectual Property Disclosure	15
7 Definitions	15
8 FIX Orchestra	15
8.1 Project milestones	15
8.1.1 Version 1.1 RC2 deliverables	15
8.2 Plan for EP production	16
8.3 Tool Migration	16
Appendix A - Usage Examples	16
Appendix B – Compliance Strategy	16

Document History

Revision	Date	Author	Revision Comments
V0.1	November 12, 2024	Hanno Klein, GTC	Initial draft
V0.2	November 26, 2024	Hanno Klein, GTC	Fixed typos identified during GTC presentation.

1 Introduction

Orchestra was conceived as **machine-readable rules of engagement** between counterparties for FIX and non-FIX interfaces. As such, it is a standard for exchange of metadata about the behavior of electronic interface applications. Orchestra is intended to cut time to onboard counterparties and improve accuracy of implementations.

1.1 Roadmap

Orchestra version 1.0 Technical Standard is deemed usable as-is, and firms are successfully building systems around it. However, several technical improvements have been proposed to increase flexibility and fix minor defects. Therefore, version 1.1 Release Candidate 1 (RC1) was proposed and published in November 2023 (<https://forum.fixtrading.org/t/public-comment-period-orchestra-version-1-1-rc1/17408>). A second release candidate is now proposed to build on RC1 and adds capabilities that belong together into a single version, i.e. version 1.1.

As always, this release candidate will be circulated for a 90-day public review. If more changes are proposed, subsequent release candidates will be published. When the working group is satisfied with the specification, a Beta Version, and ultimately, a final Technical Standard will be published.

Please see <https://www.fixtrading.org/packages/technical-standard-proposal-process/> for more details.

1.2 Authors

Name	Affiliation	Contact	Role
Hanno Klein	FIX Technical Director	hanno.klein@fixtrading.org	GTC co-chair EMEA

2 Requirements

2.1 Business Requirements

This section describes enhancements or changes to the repository schema of the Orchestra standard since Release Candidate 1 of version 1.1 was published. The issues are recorded in GitHub at

- A – <https://github.com/FIXTradingCommunity/fix-orchestra/issues> or
- B – <https://github.com/FIXTradingCommunity/fix-orchestra-spec/issues>.

The GitHub issue numbers are shown in the header of the sections below with "A-" or "B-" as prefix.

2.1.1 Scenario Relationships (#A-228)

Scenarios are one of the key features of Orchestra. There is a strong requirement to clearly define relationships between scenarios. Version 1.0 does not provide this, i.e. any relationship between scenarios of an element type needs to be defined outside of the schema.

It is proposed to support a reference from one scenario to another and to define that the referencing scenario is equal to or a subset of the referenced scenario in terms of the elements in each of the two scenarios. It is possible to change element attributes such as the description or presence rules, but it is not possible to add an element to a referencing scenario that is not also in the referenced scenario.

ISO 20022 uses this principle for its message variants¹ whereby a variant of an ISO 20022 message may only restrict its standard version. Orchestra extends this concept by making it equally applicable to groups, components, and code sets. ISO 20022 supports the latter by suffixing version numbers to business components for the definition of logical message components.

It is proposed to add a new attribute group `scenarioRefGrp` and include it into the definition of messages, groups, components, and code sets. The attribute group requires a reference to another scenario of the same element type by scenario identifier and optionally supports a reference by name. The choice of an attribute group allows to add further characteristics of scenario relationships in a future version.

```
<xs:attributeGroup name="scenarioRefGrp">
  <xs:annotation>
    <xs:documentation>A reference to a scenario by its key identifiers. There are
      no defaults as scenario references are optional.</xs:documentation>
    </xs:annotation>
    <xs:attribute name="scenarioRefId" type="fixr:id_t" use="required">
      <xs:annotation>
        <xs:documentation>Unique identifier of a scenario.</xs:documentation>
      </xs:annotation>
    </xs:attribute>
    <xs:attribute name="scenarioRef" type="fixr:Name_t">
      <xs:annotation>
        <xs:documentation>Name of a scenario. The name is optional as part of a
          reference and only for convenience.</xs:documentation>
        </xs:annotation>
      </xs:attribute>
    </xs:attributeGroup>
```

¹ <https://www.iso20022.org/catalogue-messages/additional-content-messages/variants>

2.1.2 Elaborate on message mapping to scenarios (#B-52)

Section 3.11.1.1 is a non-normative section describing message mappings to scenarios. RC1 added a `<when>` element, but the usage has not been described in the specification. It is proposed to change the existing text as follows:

The task of mapping an actual received message to a scenario declaration in Orchestra is left to implementations. The first level of matching may be on the `msgType` attribute, e.g. by comparing it with the `FIX MsgType(35)` field. However, that message type may have several scenarios. The `<when>` element as part of a `<message>` element can be used to define an expression that references specific fields in the received message.

2.1.3 Update recommendation for Dublin Core terms (#B-46)

Section 3.2.1 describes a subset of DC terms (ISO 15836) that are relevant in the context of electronic interfaces. The proposal is to update and extend this section recommending specific DC terms to be used and their semantics. Some of the DC terms are specific to document generation. RC1 only recommends a single set of terms (title, publisher, date issued, and rights) and does not identify a context for their relevance.

Recommended elements for an Orchestra spec:

- **title**: Should be a required element. If missing, should default to using the repository name.
- **creator**: The recommended role should be creator not publisher, as not all specs are published (e.g. private specs)
- **created**: The recommended date should be created not issued, as not all specs reach issuance in their lifecycle
- **conformsTo**: Tools processing Orchestra files need to know which version of the technical spec the XML conforms to, hence `conformsTo` should be a recommended element. If missing, should default to the latest official version (currently Orchestra v1.0).
- **source**: URL to another Orchestra file. Should be a recommended element if a reference file was used to create the specification. This helps to preserve provenance, and allows validation across specifications e.g. if a scenario variant in a derived spec is valid with respect the reference specification.

Additional elements recommended to support document generation:

- **alternative** - Subtitle for the resource.
- **subject** - List of keywords used for search and categorisation.
- **abstract** - A summary of the resource.
- **issued** - If missing, default to created date.
- **dateAccepted** - if missing, default to issued date. If issued date is also missing, default to created date.
- **contributor** - Secondary author, in addition to the creator.
- **publisher** - Distributor of the Orchestra file or document. If missing, default to creator.

Additional recommended elements:

- **rights** - optional, identifies the copyright holder of the *Orchestra file*
- **licence** - optional, URL to a licence file
- **isFormatOf** - URL to original resource that the Orchestra spec is based on (e.g. a PDF spec published by an Exchange).

2.1.4 Add and define documentation purposes (#A-229)

The schema has an optional attribute `purpose` for documentation elements and already defines a few standard purposes (SYNOPSIS, ELABORATION, EXAMPLE, DISPLAY), without providing any definition in plain English. A user may add to the list as it is extensible. It is proposed to add a number of standard purposes and to provide a definition for the existing and new purposes. The objective is to harmonize the use of these purposes whilst keeping the ability for the user to add own purposes for use cases that are not covered by the standard purposes.

- **SYNOPSIS** – Brief summary of the element, typically highlighting its key function or purpose, restricted to one paragraph for conciseness.
- **ELABORATION** – Detailed explanation of the element, clarifying its usage, functionality, or background.
- **EXAMPLE** – Sample or illustration demonstrating how the element is used in practice.
- **DISPLAY** – For UI when different from canonical name; may have multi-language displays.
- **CAPTION** – Descriptive label or title for the element, may be used for tables, figures, headings or brief annotations.
- **TOOLTIP** – Short message or hint that appears when hovering over the element, usually explaining its function or use.
- **DEFINITION** – Precise and formal explanation of the element, restricted to one sentence in length to ensure brevity.

2.1.5 Increase maximum length of msgType attribute (A-#208)

The schema limits the length of this attribute to 2, which is specific to FIX. Non-FIX protocols may have longer acronyms, e.g. SEC-CAT uses 5 characters to identify events and related messages, e.g. "MENO" for "New Order" and "MENOS" for "New Order Supplement".

It is proposed to maintain a limit on the attribute but to increase it to accommodate non-FIX protocols. It should remain a very short, alphanumeric identifier of a message.

```
<xs:simpleType name="MsgType_t">
  <xs:restriction base="xs:string">
    <xs:minLength value="1"/>
    <xs:maxLength value="6"/>
  </xs:restriction>
</xs:simpleType>
```


2.2 Technical Requirements for Repository Schema

This section describes enhancements or changes to the repository schema of the Orchestra standard since the v1.0 Technical Standard was published. The issues are recorded in GitHub at

- A – <https://github.com/FIXTradingCommunity/fix-orchestra/issues> or
- B – <https://github.com/FIXTradingCommunity/fix-orchestra-spec/issues>.

The GitHub issue numbers are shown in the header of the sections below with "A-" or "B-" as prefix.

2.2.1 Correct reference to mapped datatype element (B-#43)

Section 3.6.3 *Datatype mappings* has an incorrect reference to an XML element. Change as follows:

FROM: The `standard` attribute of `<datatype>` tells which type system the mapping is for.

TO: The `standard` attribute of `<mappedDatatype>` tells which type system the mapping is for.

URL: <https://github.com/FIXTradingCommunity/fix-orchestra-spec/pull/43>

2.2.2 Allow names in correlation and assignment references (A-#187)

Optional names are supported since RC1 for references to groups, components, and fields. It is proposed to also support this for correlation and assignment references used in message responses, for example:

```
<fixr:response name="orderAck">
  <fixr:messageRef id="9" name="ExecutionReport" msgType="8"
    implMaxOccurs="1" implMinOccurs="1">
    <fixr:identifiers>
      <fixr:correlate id="11" name="C1OrdID"/>
      <fixr:correlate id="2422" name="OrderRequestID"/>
      <fixr:assign id="37" name="OrderID"/>
    </fixr:identifiers>
  </fixr:messageRef>
</fixr:response>
```

2.2.3 Remove attribute applVerID (A-#186)

Orchestra has a number of attributes for the `<repository>` element. One of them (`applVerId`) is for the *application* version (not to be confused with the *repository* version attribute `version`). This corresponds to the FIX field `ApplVerID(1128)`, which is specific to FIX versions. It is proposed to remove this attribute. An alternative is to use a DC term, e.g. *hasVersion* to reference the application using the given Orchestra schema.

2.2.4 Remove unused attribute group for messages (A-#189)

The attribute group `messageAttribGrp` is defined in the schema file `repositoryTypes.xsd` but not used anywhere.

```
<xs:attributeGroup name="messageAttribGrp">
  <xs:attribute name="msgType" type="fixr:MsgType_t"/>
  <xs:attribute name="category" type="fixr:Name_t"/>
  <xs:attribute name="abbrName" type="fixr:Name_t"/>
</xs:attributeGroup>
```

```
<xs:attribute name="rendering" type="xs:string">
  <xs:annotation>
    <xs:documentation>A hint to processes about how to interpret the
      element. Not validated</xs:documentation>
  </xs:annotation>
</xs:attribute>
</xs:attributeGroup>
```

The attributes `msgType` and `category` are part of the complex type `messageType`. The attribute `abbrName` is part of the attribute group `oidGrp` inside the complex type `messageType`. Hence all but the attribute `rendering` are already available for messages in RC1. It is proposed to add the attribute `rendering` to the message type definition as follows:

```
<xs:complexType name="messageType">
  <xs:sequence>
    ...
  </xs:sequence>
  <xs:attribute name="msgType" type="fixr:MsgType_t"/>
  <xs:attribute name="category" type="fixr:Name_t"/>
  <xs:attributeGroup ref="fixr:oidGrp"/>
  <xs:attributeGroup ref="fixr:entityAttribGrp"/>
  <xs:attribute name="flow" type="fixr:Name_t"/>
  <xs:attribute name="rendering" type="xs:string">
    <xs:annotation>
      <xs:documentation>A hint to processes about how to interpret the
        element. Not validated.</xs:documentation>
    </xs:annotation>
  </xs:attribute>
</xs:complexType>
```

2.2.5 Remove unused simple types (#A-195)

The following simple types are defined in the schema but are not used elsewhere in the schema:

- `CategoryComponentType_t`
- `CompID_t`
- `GroupName_t`
- `IncludeFile_t`

They seem to have been carried over from the schema for the Unified Repository without being used in the Orchestra schema. It is proposed to remove their definitions.

2.2.6 Move datatype definition to repositoryTypes.xsd (#A-190)

The *repository.xsd* file, defines various elements for the root level of a schema, like *Messages*, *Groups*, *Fields*, *Datatypes*, etc. The definition of their individual elements is provided in the *repositoryTypes.xsd* file. There is no technical reasons for this split of definition. However, it the *Datatype* element definition does not follow this convention. It is proposed to move the *Datatype* element to the *repositoryTypes.xsd* file.

```
<xs:element name="datatype">
  <xs:complexType>
    ...
  </xs:complexType>
</xs:element>
```

2.2.7 Correct cardinality restriction of groups (#A-203)

According to the specification, a group reference should inherit the minimum (`implMinOccurs`) and maximum (`implMaxOccurs`) occurrences specified in the group definition:

Limits of the size of a particular instance of a repeating group may be overridden by setting `implMinOccurs` and `implMaxOccurs` attributes on the `<groupRef>` element.

However, the current definition overrides the upper limit with the value set in the reference itself, contradicting the specification. The problem is the presence of a default value defined in the `implMaxOccurs` attribute of `groupRefType`:

```
<xs:attribute name="implMaxOccurs" type="fixr:unboundedIntType" default="unbounded"/>
```

This behavior prevents users from defining the maximum occurrences (`implMaxOccurs`) only in the group definition. They must specify it in all references as well to avoid being overridden by the default on the group reference level. It is proposed to remove the default attribute for `implMaxOccurs` on the group level as follows:

```
<xs:attribute name="implMinOccurs" type="xs:nonNegativeInteger">
  <xs:annotation>
    <xs:documentation>Lower bound of group instances (numInGroup)</xs:documentation>
  </xs:annotation>
</xs:attribute>
<xs:attribute name="implMaxOccurs" type="fixr:unboundedIntType" default="unbounded">
  <xs:annotation>
    <xs:documentation>Upper bound of group instances (numInGroup)</xs:documentation>
  </xs:annotation>
</xs:attribute>
```

The definitions on the group reference level remain unchanged (defaulting to unbounded for the maximum value) and are as follows:

```
<xs:complexType name="groupRefType">
  <xs:complexContent>
    <xs:extension base="fixr:componentRefType">
      <xs:attribute name="implMinOccurs" type="xs:nonNegativeInteger"/>
      <xs:attribute name="implMaxOccurs" type="fixr:unboundedIntType"
        default="unbounded"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

2.2.8 Define attribute `changeType` (#B-42)

The attribute `changeType` is part of the attribute group entity `attribGrp`, which is more or less part of every element and element reference. It has not been defined in the spec and it is proposed to change the existing text as follows:

*Most message elements in the schema support a complete history of creation, **latest** change, replacement, and potentially deprecation with support of attribute group `entityAttribGrp`. Each historical event should be qualified by its protocol version and may be qualified by its release. This is an integer value, e.g. a FIX extension pack (EP) number that can be used to increase the granularity of the version string, e.g. to identify patches. **The latest change may be qualified with the `changeType` attribute as being definitional or only editorial (e.g. adding or updating a `documentation` element of an annotation).***

2.2.9 Make selected section and category attributes required (#A-191)

The element `sectionType` and `categoryType` doesn't have any required attributes. Visualization tools like [FIXimate](#) use section and category for message navigation. It is proposed to make the name required for a section and the name and section required for a category.

```
<xs:complexType name="categoryType">
  <xs:sequence>
    <xs:element name="annotation" type="fixr:annotation" minOccurs="0"/>
  </xs:sequence>
  <xs:attribute name="name" type="fixr:Name_t" use="required"/>
  <xs:attribute name="FIXMLFileName" type="fixr:Name_t"/>
  <xs:attribute name="componentType" type="fixr:CatComponentType_t"/>
  <xs:attribute name="section" type="fixr:Name_t" use="required"/>
  <xs:attribute name="includeFile" type="fixr:CatIncludeFile_t"/>
  <xs:attributeGroup ref="fixr:entityAttribGrp"/>
</xs:complexType>

<xs:complexType name="sectionType">
  <xs:sequence>
    <xs:element name="annotation" type="fixr:annotation" minOccurs="0"/>
  </xs:sequence>
  <xs:attribute name="name" type="fixr:Name_t" use="required"/>
  <xs:attribute name="displayOrder" type="xs:int"/>
  <xs:attribute name="FIXMLFileName" type="fixr:Name_t"/>
  <xs:attributeGroup ref="fixr:entityAttribGrp"/>
</xs:complexType>
```

2.2.10 Mark section of datatype mapping as non-normative (#B-47)

The following section 3.6.3 of RC1 related to datatype mappings is non-normative and has not been defined as such.

The following paragraph is non-normative.

Rather than creating numerous one-off mappings to other type systems, is it likely more efficient to map each to ISO 11404 once, and then compare mappings in an associative model to identify the commonalities. (The XML schema standard claims to derive its datatypes from ISO 11404, but mapping to the generic standard is more precise and comprehensive than filtering it through the XML interpretation.)

2.3 Technical Requirements for Interfaces Schema

This section describes enhancements or changes to the interfaces schema of the Orchestra standard since the v1.0 Technical Standard was published. The issues are recorded in GitHub at

- A – <https://github.com/FIXTradingCommunity/fix-orchestra/issues> or
- B – <https://github.com/FIXTradingCommunity/fix-orchestra-spec/issues>.

The GitHub issue numbers are shown in the header of the sections below with "A-" or "B-" as prefix.

THERE ARE NO NEW REQUIREMENTS THAT HAVE BEEN SUBMITTED FOR RC2.

2.4 Open issues

This section lists open issues that have dropped from Orchestra v1.1. The remainder of open issues for the schema or the specification has been deferred to the next Release Candidate.

The following open issues are from <https://github.com/FIXTradingCommunity/fix-orchestra>

- Missing appInfo FIXML schema
(<https://github.com/FIXTradingCommunity/fix-orchestra/issues/179>)

3 Issues and Discussion Points

NONE

4 References

Authors should list references used in creating the technical standard proposal

Reference	Version	Relevance	Normative
None			

5 Relevant and Related Standards

Related Standard	Version	Reference location	Relationship	Normative
Dublin Core XML Schemas	2008-02-11	http://dublincore.org/schemas/xmls/	Dependency	Yes
XML Schema for FIX	2016		Technical guide	Yes
XML Schema	2012	https://www.w3.org/TR/xmlschema11-1/	Dependency	Yes
Namespaces	2006	https://www.w3.org/TR/xml-names11/	Dependency	Yes

6 Intellectual Property Disclosure

Authors should provide a list of any intellectual property

Related Intellection Property	Type of IP (copyright, patent)	IP Owner	Relationship to proposed standard

7 Definitions

Term	Definition
Pedigree	The recorded history of an artifact
Provenance	A record of ownership of an artifact

8 FIX Orchestra

8.1 Project milestones

Since Orchestra has many facets, features may be delivered in several release candidates rather than attempting a big-bang approach.

8.1.1 Version 1.1 RC2 deliverables

These artifacts will be delivered as v1.1 RC2:

- The technical specification is a separate document “Orchestra Standard – Technical Specification”. The document will be displayed in the download page for Orchestra on the FIX Trading Community website (<https://www.fixtrading.org/standards/fix-orchestra-standard/>). The document source in markdown is available in the GitHub project [FIXTradingCommunity/fix-orchestra-spec](#).

These resources have been published in the GitHub project [FIXTradingCommunity/fix-orchestra](#):

- XML schema (XSD) for repository of message definitions and workflow plus documentation of the schema
- XML schema (XSD) for interfaces and session configuration plus documentation of the schema

8.2 Plan for EP production

In addition to standard development, a plan will be created to migrate the building of Extension Packs from Basic/Unified Repository to Orchestra. In addition to modernizing build facilities, this plan will need to account for the tools that consume Repository, including FIXimate, FIXML schema generation, and so forth. The tools to generate FIXimate, FIXML schema and Unified Repository have been adapted to use an Orchestra XML v1.0 file as source.

8.3 Tool Migration

Tools that have been coded to Orchestra v1.0 will need to be updated to v1.1 whenever FIX Latest moves to Orchestra v1.1.

- FIXimate4 generator
- Log2orchestra
- Playlist
- Tablature
- FIXML schema generator
- SBE schema generator
- Unified Repository

Appendix A - Usage Examples

This is a required section where the sub-committee or working group can provide whole or fragments of example FIX messages with actual or dummy data. These examples are useful for illustrating usage or rules specific to the business domain covered in the proposal.

These example Orchestra files are posted in GitHub but are so far only applicable to Orchestra v1.0.

Example order entry file developed by MilleniumIT

`fix-orchestra/repository/src/test/resources/examples/`

Sample interface file

<https://github.com/FIXTradingCommunity/fix-orchestra/blob/master/interfaces/src/test/resources/SampleInterfaces.xml>

A non-FIX exchange API interpreted in Orchestra

<https://github.com/FIXTradingCommunity/orchestrations/tree/master/NYSE%20Pillar>

Appendix B – Compliance Strategy

The technical standard must include some plan for measuring compliance with the standard. This will either be test suites or a validation tool (such as an XML Schema document as an example).

The first level of compliance will be provided by existing XML tools that verify conformity of a file to its schema. A test is provided to validate that isolated DSL expressions conform to the grammar. However, a comprehensive compliance test is not being developed by FIX.