BEA WebLogic Adapter for FIX®

User Guide

Version 8.1.2
Document Revised: January 2004

Copyright

Restricted Rights Legend

Trademarks or Service Marks

Contents

About This Document

1. Introducing the BEA WebLogic Adapter for FIX

2. Generating Schemas for FIX Integration Objects

3. Defining Application Views for FIX

A. Supported Messages

Index

About This Document

This document describes how to use the BEA WebLogic Adapter for FIX. This document is organized as follows:

- Chapter 1, "Introducing the BEA WebLogic Adapter for FIX," describes the adapter, how it relates to both FIX business objects and WebLogic Integration.
- Chapter 2, "Generating Schemas for FIX Integration Objects," describes how to generate schemas for your FIX business objects.
- Chapter 3, "Defining Application Views for FIX," describes application views and how to use them to configure events and services.
- Appendix A, "Supported Messages," describes the FIX messages supported by the adapter.

Who Should Read This Documentation

This document is intended for the following members of an integration team:

- Integration Specialists—Lead the integration design effort. Integration specialists have expertise in defining the business and technical requirements of integration projects, and in designing integration solutions that implement specific features of WebLogic Integration. The skills of integration specialists include business and technical analysis, architecture design, project management, and WebLogic Integration product knowledge.
- Technical Analysts—Provide expertise in an organization's information technology infrastructure, including telecommunications, operating systems, applications, data

repositories, future technologies, and IT organizations. The skills of technical analysts include technical analysis, application design, and information systems knowledge.

- Enterprise Information System (EIS) Specialists—Provide domain expertise in the systems that are being integrated using WebLogic Integration adapters. The skills of EIS specialists include technical analysis and application integration design.
- System Administrators—Provide in-depth technical and operational knowledge about databases and applications deployed in an organization. The skills of system administrators include capacity and load analysis, performance analysis and tuning, deployment topologies, and support planning.

Additional Information

To learn more about the software components associated with the adapter, see the following documents:

- *BEA WebLogic Adapter for FIX Release Notes*

  http://edocs.bea.com/wladapters/fix/docs812/pdf/relnotes.pdf
- *BEA WebLogic Adapter for FIX Installation and Configuration Guide*

  http://edocs.bea.com/wladapters/fix/docs812/pdf/install.pdf
- *Introduction to the BEA WebLogic Adapters*

  http://edocs.bea.com/wladapters/docs81/pdf/intro.pdf
- BEA WebLogic Adapters 8.1 Dev2Dev Product Documentation

  http://dev2dev.bea.com/products/wladapters/index.jsp
- Application Integration documentation

  http://edocs.bea.com/wli/docs81/aiover/index.html

  http://edocs.bea.com/wli/docs81/aiuser/index.html
- BEA WebLogic Integration documentation

  http://edocs.bea.com/wli/docs81/index.html
- BEA WebLogic Platform documentation

  http://edocs.bea.com/platform/docs81/index.html
- FIX documentation

  http://www.fixprotocol.org

How to Use This Document

This document is designed to be used in conjunction with *Using the Application Integration Design Console*, available at the following URL:

`http://edocs.bea.com/wli/docs81/aiuser/index.html`

Using the Application Integration Design Console describes, in detail, the process of defining an application view, which is a key part of making an adapter available to process designers and other users. What *Using the Application Integration Design Console* does *not* cover is the specific information about Adapter for FIX that you need to supply to complete the application view definition. You will find that information in this document.

At each point in *Using the Application Integration Design Console* where you need to refer to this document, you will see a note that directs you to a section in your adapter user guide, with a link to the edocs page for adapters. The following roadmap illustration shows where you need to refer from *Using the Application Integration Design Console* to this document.

Figure 1 Information Interlock with *Using the Application Integration Design Console*

Contact Us!

Your feedback on the BEA WebLogic Adapter for FIX documentation is important to us. Send us e-mail at **docsupport@bea.com** if you have questions or comments. Your comments will be reviewed directly by the BEA professionals who create and update the BEA WebLogic Adapter for FIX documentation.

In your e-mail message, please indicate that you are using the documentation for BEA WebLogic Adapter for FIX and the version of the documentation.

If you have any questions about this version of BEA WebLogic Adapter for FIX, or if you have problems using the BEA WebLogic Adapter for FIX, contact BEA Customer Support through BEA WebSUPPORT at **www.bea.com**. You can also contact Customer Support by using the contact information provided on the Customer Support Card which is included in the product package.

When contacting Customer Support, be prepared to provide the following information:

- Your name, e-mail address, phone number, and fax number
- Your company name and company address
- Your machine type
- The name and version of the product you are using
- A description of the problem and the content of pertinent error messages

Documentation Conventions

The following documentation conventions are used throughout this document.

Convention	Item
boldface text	Indicates terms defined in the glossary.
Ctrl+Tab	Indicates that you must press two or more keys simultaneously.
italics	Indicates emphasis or book titles.

`monospace text`	Indicates code samples, commands and their options, data structures and their members, data types, directories, and file names and their extensions. Monospace text also indicates text that you must enter from the keyboard.
	Examples:
	`#include <iostream.h> void main ( ) the pointer psz`
	`chmod u+w *`
	`\tux\data\ap`
	`.doc`
	`tux.doc`
	`BITMAP`
	`float`

`monospace boldface text`	Identifies significant words in code. *Example*: `void` **`commit`** `( )`
`monospace italic text`	Identifies variables in code. *Example*: `String` *`expr`*
UPPERCASE TEXT	Indicates device names, environment variables, and logical operators. *Example*s: LPT1 SIGNON OR
`{ }`	Indicates a set of choices in a syntax line. The braces themselves should never be typed.

Convention	Item
`[ ]`	Indicates optional items in a syntax line. The brackets themselves should never be typed. *Example*: `buildobjclient [-v] [-o name ] [-f file-list]... [-l file-list]...`
|	Separates mutually exclusive choices in a syntax line. The symbol itself should never be typed.
. . .	Indicates one of the following in a command line: • That an argument can be repeated several times in a command line • That the statement omits additional optional arguments • That you can enter additional parameters, values, or other information The ellipsis itself should never be typed. *Example*: `buildobjclient [-v] [-o name ] [-f file-list]... [-l file-list]...`
. . .	Indicates the omission of items from a code example or from a syntax line. The vertical ellipsis itself should never be typed.

CHAPTER 1

Introducing the BEA WebLogic Adapter for FIX

This section introduces the BEA WebLogic Adapter for FIX and describes how the adapter enables integration with FIX business objects and WebLogic Integration.

It includes the following topics:

- About the BEA WebLogic Adapter for FIX
- About the BEA WebLogic Adapter for FIX and Business Processes
- Getting Started With the Adapter for FIX

About the BEA WebLogic Adapter for FIX

The BEA WebLogic Adapter for FIX connects to your FIX system so that you can easily use your FIX data and functions within your business processes. The adapter provides scalable, reliable, and secure access to your FIX system.

This section includes the following topics:

- Supported FIX Operations for Application Integration
- Supported Services
- Supported Events
- Benefits of the Adapter for FIX

Supported FIX Operations for Application Integration

The Adapter for FIX supports synchronous and asynchronous, bi-directional message interactions for FIX messages.

It provides integration with the following FIX operations:

- Support for FIX messages using XML to handle both services and events
- Support for multiple versions of the protocol (FIX 4.0, 4.1, 4.2, 4.3, and 4.4) across multiple sessions on a single instance. The BEA WebLogic Adapter for FIX fully supports the full 39 message types defined in FIX 4.2 specification that support pre-trade, trade, post-trade, and clearing/settlement processes in the securities industry.
- Support for multiple simultaneous FIX sessions
- Dynamic formatting and transformation of FIX messages into FIXML for internal processing. This enables organizations to use FIXML for communications with internal systems and FIX messages to communicate across networks with trading counter-parties. Even complex workflows for straight-through processing (STP), a major initiative in the securities industry, are simply managed by teaming the BEA WebLogic Adapter for FIX with other components from the BEA WebLogic Integration platform.
- Support for pre- and post-encryption of messages using DES and PGP/DES/MD5

Supported Services

The Adapter for FIX supports one type of service, a FIX service. Using this service, the adapter writes a file to your file system.

Supported Events

The Adapter for FIX supports two event models:

- Buyside events. For this event, the adapter picks up a file and passes it to an event variable within a business process. This event type can also receive a FIX message by listening at a specified host and port number.
- Sellside events. For this event, the adapter picks up a file and passes it to an event variable within a business process.

Benefits of the Adapter for FIX

The combination of the adapter and WebLogic Integration supplies everything you need to integrate your workflows and enterprise applications with your FIX system. The Adapter for FIX provides these benefits:

- Integration can be achieved without custom coding.
- Business processes can be started by events generated by FIX.
- Business processes can request and receive data from your FIX system using services.
- Adapter events and services are standards-based. The adapter services and events provide extensions to the *J2EE Connector Architecture* (JCA) version 1.0 from Sun Microsystems, Inc. For more information, see the Sun JCA page at the following URL:

  `http://java.sun.com/j2ee/connector/`
- The adapter and WebLogic Integration solution is scalable. The BEA WebLogic Platform provides clustering, load balancing, and resource pooling for a scalable solution. For more information about scalability, see the following URL:

  `http://edocs.bea.com/wls/docs81/cluster/index.html`
- The adapter and WebLogic Integration solution benefits from the fault-tolerant features of the BEA WebLogic Platform. For more information about high availability, see the following URL:

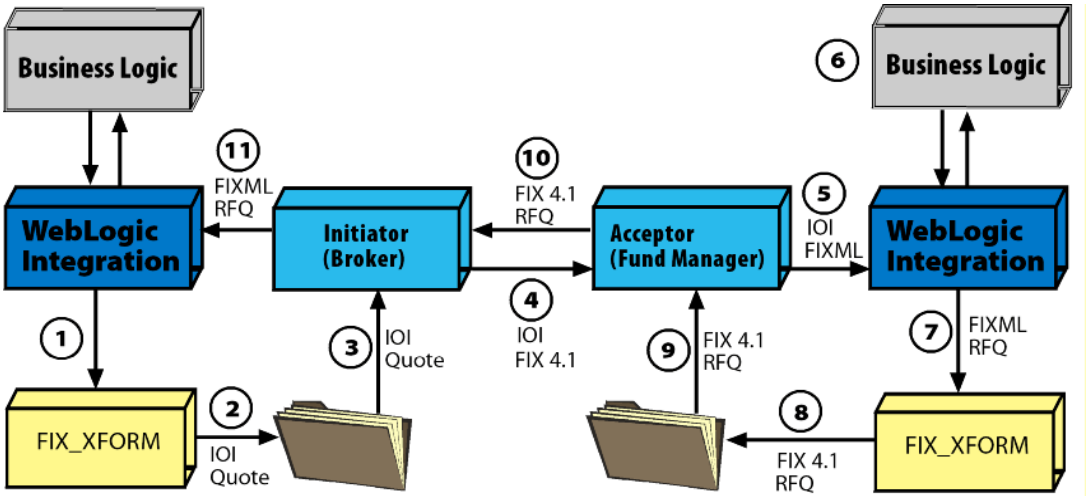
  `http://edocs.bea.com/wli/docs81/deploy/index.html`
- The adapter and WebLogic Integration solution is secure, using the security features of the BEA WebLogic Platform and the security of your FIX system. For more information about security, see the following URL:

  `http://edocs.bea.com/wls/docs81/secintro/index.html`

About the BEA WebLogic Adapter for FIX and Business Processes

BEA WebLogic Integration when combined with the BEA WebLogic Adapter for FIX allows the implementation of workflow processes for pre-trade and post-trade business processes. To facilitate this, the BEA WebLogic Adapter for FIX provides transformation services for FIX messages.

The following diagram assumes that both the Initiator (Seller) and the Acceptor (buyer) have WebLogic environments with the BEA WebLogic Adapter for FIX installed.



The diagram illustrates the exchange of FIX messages as follows:

1. An indication of interest (IOI) is spawned on the Broker side by business logic established in the BEA WebLogic Integration environment.
2. The IOI is transformed into FIX protocol and placed in a file repository.
3. The FIX IOI is picked up from the repository
4. The FIX IOI is sent to the Fund Manager
5. The FIX message is converted to FIXML by the adapter and passed to the BEA WebLogic Integration environment.
6. The Fund Manager can apply business logic in its WebLogic environment to decide whether to respond to the Broker.
7. If the decision is made to respond with a request for a quote (RFQ), the FIXML message is transformed to FIX protocol.
8. The FIX RFQ is placed in a file repository.
9. The FIX RFQ is picked up from the file repository.
10. The FIX RFQ is sent to the initiator.

11. The FIX message is transformed to FIXML by the adapter and exposed to business logic in the BEA WebLogic Integration environment.

Getting Started With the Adapter for FIX

This section gives an overview of how to get started using the BEA WebLogic Adapter for FIX within the context of an application integration solution. Integration with FIX involves the following tasks:

- Step 1: Design the Application Integration Solution
- Step 2: Determine the Required FIX Business Workflows
- Step 3: Generate Schemas for FIX Integration Objects
- Step 4: Define Application Views and Configure Services and Events
- Step 5: Integrate Your Application with Other BEA Software Components
- Step 6: Deploy the Solution to the Production Environment

Step 1: Design the Application Integration Solution

The first step is to design an application integration solution, which includes (but is not limited to) such tasks as:

- Defining the overall scope of application integration.
- Determining the business process(es) to integrate.
- Determining which WebLogic Platform components will be involved in the integration, such as web services or workflows designed in WebLogic Workshop, portals created in WebLogic Portal, and so on.
- Determining which external systems and technologies will be involved in the integration, such as FIX systems and other EISs.
- Determining which BEA WebLogic Adapters for WebLogic Integration will be required, such as the BEA WebLogic Adapter for FIX. An application integration solution can involve multiple adapters.

This step involves the expertise of business analysts, system integrators, and EIS specialists (including FIX specialists). Note that an application integration solution can be part of a larger integration solution.

Step 2: Determine the Required FIX Business Workflows

Within the larger context of an application integration project, you must determine which specific FIX messages are required for services and events to support the business processes in the application integration solution.

Factors to consider include (but are not limited to):

- Type of FIX messages used
- Transactions using FIX involved in business processes
- Whether operations are, from the adapter point of view:
  - services, which notify the FIX system with a request for action
  - events, which are notifications from the FIX system that trigger workflows

This step involves the expertise of FIX specialists, including analysts and administrators.

Step 3: Generate Schemas for FIX Integration Objects

After identifying the FIX integration objects and workflows required for the application integration solution, you must generate the XML schemas that will be used to exchange data with one or more FIX systems:

- Services require two XML schemas: one for the FIX request and another for the FIX response.
- Events require a single XML schema to handle the data sent by the FIX system.

To learn more about schemas, see Chapter 2, "Generating Schemas for FIX Integration Objects."

Step 4: Define Application Views and Configure Services and Events

After you create the schemas for your FIX services or events, you create an application view that provides an XML-based interface between WebLogic Server and a particular FIX system within your enterprise. If you are accessing multiple FIX systems, you define a separate application view for each FIX system you want to access. To provide different levels of security access (such as "guest" and "administrator"), define a separate application view for each security level.

Once you define an application view, you can configure events and services in that application view that employ the XML schemas that you created in "Step 3: Generate Schemas for FIX

Integration Objects" on page 1-6. To learn more about generating schemas, see Chapter 2, "Generating Schemas for FIX Integration Objects."

To learn more about defining application views, see Chapter 3, "Defining Application Views for FIX" in conjunction with*Using the Application Integration Design Console*, at the following URL:

`http://edocs.bea.com/wli/docs81/aiuser/index.html`

Step 5: Integrate Your Application with Other BEA Software Components

Once you have configured and published one or more application views for FIX integration, you can integrate these application views into other BEA software components, such as workflows or web services created in BEA WebLogic Workshop, or portals built with BEA WebLogic Portal.

For more information, see *Using the Application Integration Design Console*, particularly Chapter 3, "Using Application Views with Application Workflows," at the following URL:

`http://edocs.bea.com/wli/docs81/aiuser/index.html`

Step 6: Deploy the Solution to the Production Environment

After you have designed, built, and tested your application integration solution, you can deploy it into a production environment. The following list describes some of the tasks involved in deploying an application integration:

- Design the deployment.
- Deploy the required components of the BEA WebLogic Platform.
- Install and deploy the BEA WebLogic Adapter for FIX as described in *BEA WebLogic Adapter for FIX Installation and Configuration Guide*
- Deploy your application views and schemas for FIX integration.
- Verify business processes in the production environment.
- Monitor and tune the deployment.

CHAPTER 2

Generating Schemas for FIX Integration Objects

This section explains how metadata for your enterprise information system (EIS) is described, how to name a schema repository and the schema manifest, how to create a schema, and how to store directory and template files for transformations. After the metadata for your EIS is described, you can create and deploy application views using the WebLogic Application View Console. The BEA WebLogic Adapter for FIX provides schemas that match each version of FIX supported by the adapter.

This section includes the following topics:

Understanding Metadata

When you define an application view, you are creating an XML-based interface between WebLogic Integration and an enterprise information system (EIS) or application within your enterprise. The BEA Adapter for FIX is used to define an interface to applications within and outside of the enterprise. The BEA Adapter for FIX supports both a FIX client-server interface and a file-based interface. Many applications or information systems use file systems to store and share data. These files contain information required by other applications, and this information can be fed via the BEA WebLogic Adapter for FIX.

About Schemas and Repositories

You describe all the documents entering and exiting your WebLogic Integration system using W3C XML schemas. These schemas describe each event arriving to and propagating out of an event, and each request sent to and each response received from a service. There is one schema for each event and two for each service (one for the request, one for the response). The schemas are usually stored in files with an `.xsd` extension.

The BEA_FIX_8_1.ear file automatically generates repository directories and components.

Use the WebLogic Integration Application View Console to access events and services, and to assign a schema to each event, request, and response. Assign each application view to a schema repository; several application views can be assigned to the same repository.

BEA WebLogic Adapters all make use of a schema repository to store their schema information and present it to the WebLogic Application View Console. The schema repository is a directory containing:

- A manifest file that describes the event and service schemas
- The corresponding schema descriptions

To work with schemas, you must know how to:

- Name a schema repository
- Create a manifest
- Create a schema

Naming a Schema Repository

The schema repository has a three-part naming convention:

session_base_directory`\`*adapter*`\`*connection_name*

- *session_base_directory* is the schema's session base path, which represents a folder under which multiple sessions of schemas may be held.
- *adapter* is the type of adapter (for example, FIX or SAP).
- *connection_name* is a name representing a particular instance of the adapter type.

For example, if the session base path is `/usr/opt/bea/bse`, the adapter type is FIX, and the connection name is FIXDev, then the schema repository is the directory:

```
/usr/opt/bea/bse/FIX/FIXDev
```

About the Repository Manifest

Each schema repository has a manifest that describes the repository and its schemas. This repository manifest is stored as an XML file named `manifest.xml`.

The following is an example of a sample manifest file showing relationships between events and services and their schemas.

The manifest file relates documents (through their schemas) to services and events. The manifest exposes schema references to the event relating the required document (via the root tag) to the corresponding schema. Schemas and manifests are stored in the same directory, the repository root of the EIS. The following is an example of the manifest file with a description of the elements.

Listing 2-1 Sample Manifest File

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<manifest>
       <connection/>
       <schemaref name="version_4_1"
              <request root="FIXMLMessage" file="FIXML41.xsd"/>
              <response root="emitStatus" file="emitStatus.xsd"/>
              <event root="FIXMLMessage" file="FIXML41.xsd"/>
       </schemaref>
       <schemaref name="version_4_2">
              <request root="FIXMLMessage" file="FIXML42.xsd"/>
              <response root="emitStatus" file="emitStatus.xsd"/>
              <event root="FIXMLMessage" file="FIXML42.xsd"/>
       </schemaref>
       <schemaref name="version_4_3">
              <request root="FIXMLMessage" file="FIXML43.xsd"/>
              <response root="emitStatus" file="emitStatus.xsd"/>
              <event root="FIXMLMessage" file="FIXML43.xsd"/>
       </schemaref>
</manifest>
```

The manifest has a connection section (which is not used by the BEA WebLogic Adapter for FIX) and a schema reference section, named schemaref. The schema reference name is displayed in the schema drop-down list on the Add Service and Add Event windows in the WebLogic Integration Application View Console. This sample manifest has three schema references or schemaref tags; one for services only, one for events only, and one for a combination of services and events. Events require only one schema, defined by the *event* tag. This relates the root tag of an XML document to a schema in the EIS repository. For services, two schemas are required: one for the document being passed to the service, represented by the `request` tag, and one for the expected *response* document received from the service operation, represented by the `response` tag.

Mapping FIX Schemas

The BEA WebLogic Adapter for FIX provides schemas that match each version of FIX supported by the adapter. The tag names used in the schemas reflect the naming conventions used in each version of the FIX standard. When choosing a schema to use for creating a FIXML message, be sure to choose the schema that matches the field definitions required for the FIXML message you are constructing.

For FIX field definitions, see the following URLs:

FIX version	URL
4.0	http://www.fixprotocol.org/specification/xml/fiximate-40/index.html
4.1	http://www.fixprotocol.org/specification/xml/fiximate-41/index.html
4.2	http://www.fixprotocol.org/specification/xml/fiximate-42/index.html
4.3	http://www.fixprotocol.org/specification/xml/fiximate-43_with20020920Errata/index.html
4.4	http://www.fixprotocol.org/specification/xml/FIXimate44/index.html

CHAPTER 3

Defining Application Views for FIX

An application view is a business-oriented interface to objects and operations within an EIS.

This section presents the following topics:

How to Use This Document

This document is designed to be used in conjunction with *Using the Application Integration Design Console*, available at the following URL:

`http://edocs.bea.com/wli/docs81/aiuser/index.html`

Using the Application Integration Design Console describes, in detail, the process of defining an application view, which is a key part of making an adapter available to process designers and other users. What *Using the Application Integration Design Console* does *not* cover is the specific information—about connections to your FIX system, as well as supported services and events—that you must supply as part of the application view definition. You will find that information in this section.

At each point in *Using the Application Integration Design Console* where you need to refer to this document, you will see a note that directs you to a section in your adapter user guide, with a link to the edocs page for adapters. The following road map illustration shows where you need to refer from *Using the Application Integration Design Console* to this document.

Figure 3-1 Information Interlock with *Using the Application Integration Design Console*

Before You Begin

Before you define an application view, make sure you have:

- Installed and deployed the adapter according to the instructions in *BEA WebLogic Adapter for FIX Installation and Configuration Guide*.
- Determined which business processes need to be supported by the application view. The required business processes determine the types of services and events you include in your application views. Therefore, you must gather information about the application's business requirements from the business analyst. Once you determine the necessary business

processes, you can define and test the appropriate services and events. For more information, see "Getting Started With the Adapter for FIX" on page 1-5.

About Application Views

An application view defines:

- Connection information for the EIS, including login information, connection settings, and so on.
- Service invocations, including the information the EIS requires for this request, as well as the request and response schemas associated with the service.
- Event notifications, including the information the EIS publishes and the event schema for inbound messages.

Typically, an application view is configured for a single business purpose and contains only the services and events required for that purpose. An EIS might have multiple application views, each defined for a different purpose.

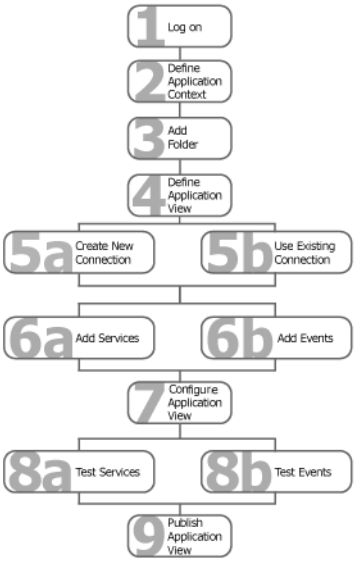
About Defining Application Views

Defining an application view is a multi-step process described in *Using the Application Integration Design Console*, available at the following URL:

`http://edocs.bea.com/wli/docs81/aiuser/index.html`

The information you enter depends on the requirements of your business process and your EIS system configuration. Figure 3-2 summarizes the procedure for defining and configuring an application view.

Figure 3-2 Process for Defining and Configuring an Application View



To define an application view:

1. Log on to the WebLogic Integration Application View Console.
2. Define the application context by selecting an existing application or specifying a new application name and root directory.

   This application will be using the events and services you define in your application view. The application view works within the context of this application.
3. Add folders as required to help you organize application views.
4. Define a new application view for your adapter.
5. Add a new connection service or select an existing one.

   If you are adding a new connection service, see "Defining Service Connection Parameters" on page 3-5 for details about FIX requirements.
6. Add the events and services for this application view.

   See the following sections for details about FIX requirements:

   – "Setting Service Properties" on page 3-6

– "Setting Event Properties" on page 3-9

7. Perform final configuration tasks.

   If you are adding an event connection, see "Defining Event Connection Parameters" on page 3-14 for details about FIX requirements.

8. Test all services and events to make sure they can properly interact with the target FIX system.

   See the following sections for details about testing:

   – "Testing Services" on page 3-16
   – "Testing a Buyside Event Using a Sellside Event" on page 3-17
   – "Testing a Buyside Event Using a FIX Client" on page 3-18
   – "Testing a Sellside Event Using a Buyside Event" on page 3-19
   – "Testing a Sellside Event Using a FIX Server" on page 3-20

9. Publish the application view to the target WebLogic Workshop application.

   This is the application you specified in step 2. Publishing the application view allows workflow developers within the target application to interact with the newly published application view using an Application View control.

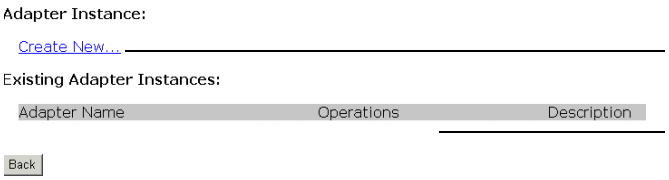
Defining Service Connection Parameters

1 2 3 4 **5** 6 7 8 9

This information applies to "Step 5A, Create a New Browsing Connection" in *Using the Application Integration Design Console*, at the following URL:

`http://edocs.bea.com/wli/docs81/aiuser/index.html`

The Select Browsing Connection page allows you to choose the type of connection factory to associate with the application view. You can select a connection factory within an existing instance of the adapter or create a connection factory within a new adapter instance.

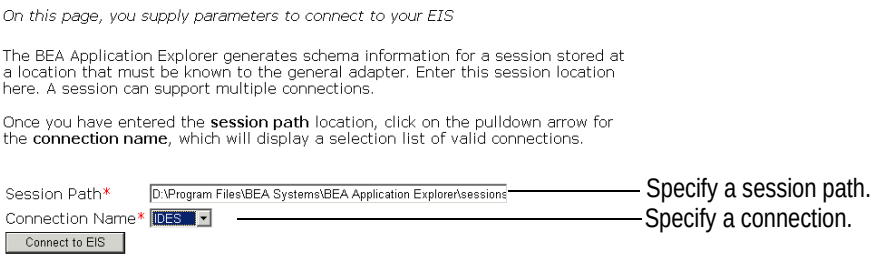


After you enter a connection name and description, you use the Configure Connection Parameters page to specify connection parameters for a connection factory.

To create a new browsing connection:

1. In the Create New Browsing Connections page, enter a connection name and description as described in *Using the Application Integration Design Console*.

   The Configure Connection Parameters page appears to allow you to configure the newly created connection factory within the new adapter instance.



   Note: A red asterisk (*) indicates that a field is required.

2. Specify a session path and connection name.

   This information enables the application view to interact with the target FIX system. You need enter this information only once per application view.

3. Click Connect to EIS.

   You return to the Create New Browsing Connections, where you can specify connection pool parameters and logging levels. For more information, see *Using the Application Integration Design Console* at the following URL:

   `http://edocs.bea.com/wli/docs81/aiuser/index.html`

Setting Service Properties

1 2 3 4 5 **6** 7 8 9

This information applies to "Step 6A, Add a Service to an Application View" in *Using the Application Integration Design Console*, at the following URL:

`http://edocs.bea.com/wli/docs81/aiuser/index.html`

The Adapter for FIX uses services to make requests of the FIX system. A service consists of both a request and a response. The Adapter for FIX supports one service:

FIX Service

1 2 3 4 5 **6** 7 8 9

This information applies to "Step 6A, Add a Service to an Application View" in *Using the Application Integration Design Console*, at the following URL:

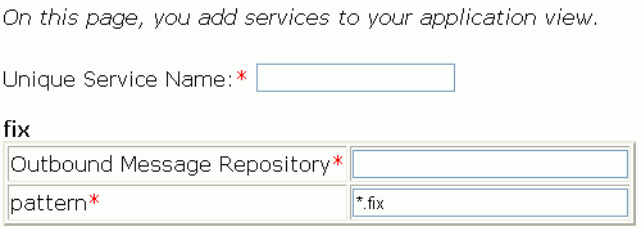
`http://edocs.bea.com/wli/docs81/aiuser/index.html`

A FIX service sends a file to a message repository, where it is retrieved by your FIX system.

To configure a FIX Service:

1. Enter a unique service name that describes the function the service performs.

   The Add Services page displays the fields required for this service type.



   Note: A red asterisk (*) indicates that a field is required.

2. Enter the following information:

Table 3-1 FIX Service Parameters

Parameter	Description
Outbound Message Repository	Repository where the outbound messages are stored
pattern	The file pattern used to store the FIX messages. A * in the file name expands to a timestamp. A # in the file name is a mask for a sequence count. Each pound symbol (#) represents an integer value. For example, File## counts up to 99 before restarting at 0. File### counts up to 999 before restarting at 0, and so on.

3. See "Common Service and Event Settings" on page 3-8 for information about selecting a schema and configuring logging and tracing.

Common Service and Event Settings

1 2 3 4 5 **6** 7 8 9

This information applies to "Step 6A, Add a Service to an Application View" in *Using the Application Integration Design Console*, at the following URL:

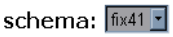
`http://edocs.bea.com/wli/docs81/aiuser/index.html`

You select a schema and select logging options the same way for all services.

To set common service settings:

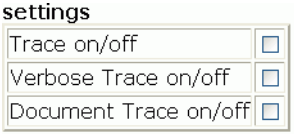
1. In the Schema list, select the schema you want to use with this service or event.

   To learn more, see Chapter 2, "Generating Schemas for FIX Integration Objects."

2. Configure logging and tracing for this service, as follows:

   Logging captures information from your adapter and writes it in a log file. Tracing displays runtime information in the console. You set the type and amount of information you wish to capture as part of the final configuration tasks. This is described in detail in *Using the Application Integration Design Console*.

   a. Select the Trace on/off check box to enable tracing for this service. Trace information appears in the runtime console.

   b. Select the Verbose Trace check box to enable additional trace information for deeper troubleshooting.

   c. Select the Document Trace check box to enable inclusion of complete incoming and outgoing documents in the trace.

3. Click Add to add the service or event.

   For more information about the next step, see *Using the Application Integration Design Console* at the following URL:

http://edocs.bea.com/wli/docs81/aiuser/index.html

Setting Event Properties

This information applies to "Step 6B, Add an Event to an Application View" in *Using the Application Integration Design Console*, at the following URL:

http://edocs.bea.com/wli/docs81/aiuser/index.html

An event defines how your application responds to events generated by FIX. The Adapter for FIX supports the following events:

- Buyside Event
- Sellside Event

Buyside Event

This information applies to "Step 6B, Add an Event to an Application View" in *Using the Application Integration Design Console*, at the following URL:

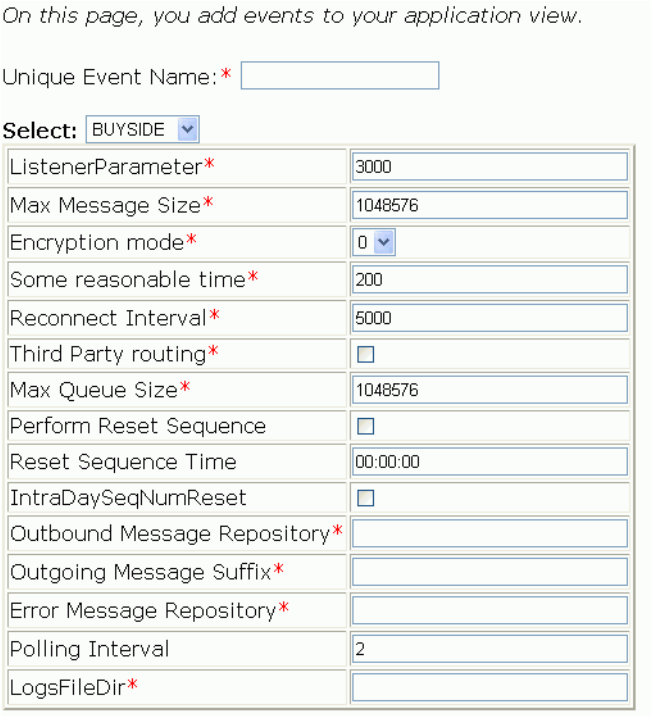
http://edocs.bea.com/wli/docs81/aiuser/index.html

In a Buyside Event, the adapter picks up a file for a buyside event, or listens on a port for the FIX message.

To configure a Buyside Event:

1. Enter a unique event name that describes the function the event performs.
2. Select BUYSIDE from the Select list.

   The Add Events page displays the fields required for this event type.



Note: A red asterisk (*) indicates that a field is required.

3. Enter the following information:

Table 3-2 Buyside Event Parameters

Parameter	Definition
ListenerParameter	TCP/IP port on which FIX initiator communicates to FIX acceptor
Max Message Size	The maximum value of allowed message size in bytes. Messages that exceed this size will be garbled
Encryption mode	The default value of encryption mode
Some reasonable time	Specifies a reasonable transmission time of FIX specification, measured in milliseconds

Table 3-2 Buyside Event Parameters

Parameter	Definition
Reconnect Interval	Specifies interval between two reconnect attempts, made at the event of connection break, measured in milliseconds
Third Party routing	Specifies whether to work in the router mode
Max Queue Size	The maximum message queue size cached in RAM, measured in bytes. It does not limit message queue size in persistent storage.
Perform Reset Sequence	Specifies whether to reset sequence number after time specified in Reset Sequence Time
Reset Sequence Time	Specifies GMT time when the FIX Engine initiates the reset of sequence numbers. Valid values: HH:MM:SS time format
IntraDaySeqNumReset	Specifies whether to reset sequence number after logout
Outbound Message Repository	Repository where outbound messages are stored. Only FIXML messages can be stored in this repository.
Outgoing Message Suffix	Suffix of the file placed in the outbound repository. The suffix should be `fix`. **Note:** There is no need to enter a period before the suffix.
Error Message Repository	Repository for storing messages that are not processed correctly
Polling Interval	Interval, in seconds, at which to check the Outbound Message Repository for messages
LogsFileDir	Specifies directory for storing session-related information

4. See "Common Service and Event Settings" on page 3-8 for information about selecting a schema and configuring logging and tracing.

Sellside Event

1 2 3 4 5 **6** 7 8 9

This information applies to "Step 6B, Add an Event to an Application View" in *Using the Application Integration Design Console*, at the following URL:

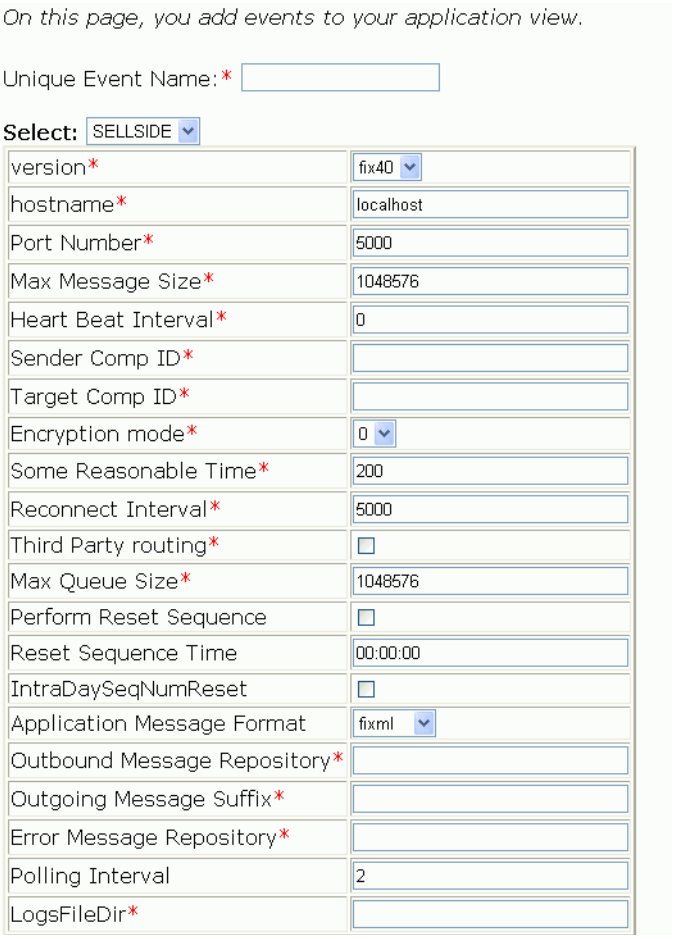
`http://edocs.bea.com/wli/docs81/aiuser/index.html`

In a Sellside Event, the adapter picks up a file for a sellside event.

To configure a Sellside Event:

1. Enter a unique event name that describes the function the event performs.
2. Select SELLSIDE from the Select list.

   The Add Events page displays the fields required for this event type.



   Note: A red asterisk (*) indicates that a field is required.

3. Enter the following information:

Table 3-3 Sellside Event Parameters

Parameter	Definition
version	Fix protocol version number in which to send FIX message
hostname	Name of the machine where the buyside event is deployed and listening
Port Number	TCP/IP port on which FIX initiator communicates to FIX acceptor
Max Message Size	The maximum value of allowed message size in bytes. Messages that exceed this size will be garbled
Heart Beat Interval	Interval between heartbeat messages sent by the FIX application, measured in milliseconds. Heartbeat messages are used to monitor the status of the communication link. These messages can also be used to identify any gaps in the sequence numbers of incoming messages. The default is 0.
Sender Comp ID	Name of the FIX initiator
Target Comp ID	Name of FIX acceptor
Encryption mode	The default value of encryption mode
Some Reasonable Time	Specifies a reasonable transmission time of FIX specification, measured in milliseconds
Reconnect Interval	Specifies interval between two reconnect attempts, made at the event of connection break, measured in milliseconds
Third Party routing	Specifies whether to work in the router mode
Max Queue Size	The maximum message queue size cached in ram, measured in bytes. It doesn't limit message queue size in persistent storage.
Perform Reset Sequence	This parameter specifies whether to reset sequence number after time specified in Reset Sequence Time
Reset Sequence Time	Specifies GMT time when the FIX Engine initiates the reset of sequence numbers. Valid values: HH:MM:SS time format
IntraDaySeqNumReset	Specifies whether to reset sequence number after Logout

Table 3-3 Sellside Event Parameters

Parameter	Definition
Application Message Format	Specifies the format of fix message delivery to the application. Valid values are fixmsg or fixml.
Outbound Message Repository	Repository where outbound messages are stored. Only FIXML messages can be stored in this repository.
Outgoing Message Suffix	Suffix of the file placed in the outbound repository. The suffix should be `fix`. **Note:** There is no need to enter a period before the suffix.
Error Message Repository	Repository for storing messages that are not processed correctly
Polling Interval	Interval, in seconds, at which to check the Outbound Message Repository for messages
LogsFileDir	Specifies directory for storing session-related information

4. See "Common Service and Event Settings" on page 3-8 for information about selecting a schema and configuring logging and tracing.

Defining Event Connection Parameters

This information applies to "Step 7, Perform Final Configuration Tasks" in *Using the Application Integration Design Console*, at the following URL:

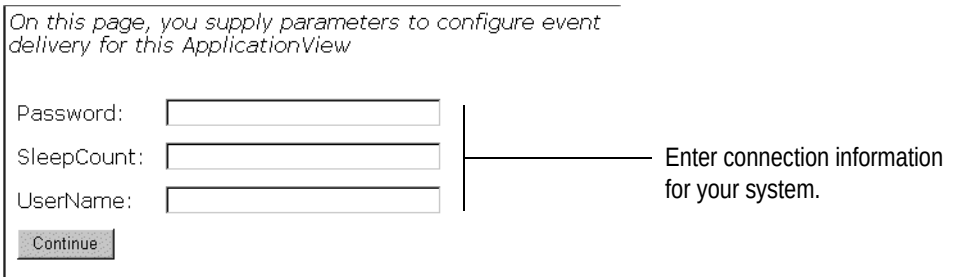
`http://edocs.bea.com/wli/docs81/aiuser/index.html`

Once you have finished adding services and events and have saved your application view, you must perform some final configuration tasks, including configuring event delivery connections, before testing the services and events. You perform these configuration tasks from the Final Configuration and Testing page.

To define event connection parameters:

1. In Connections area on the Application View Administration page, click Select/Edit.
2. In the Event Connection area, click Event to edit the default event connection.

   The Configure Event Delivery Parameters page appears.



Note: A red asterisk (*) indicates that a field is required.

3. Enter the following information:

Table 3-4 Event Connection Parameters

Parameter	Description
Password	The password for your WebLogic Server Administration Console user name
SleepCount	The number of seconds the adapter will wait between polling for events
UserName	Your WebLogic Server Administration Console user name, defined in the `startWebLogic` script

The event delivery parameters you enter on this page enable connection to your FIX system and are used when generating events. The parameters are specific to the associated adapter and are defined in the `wli-ra.xml` file within the base adapter.

4. Click Continue to save your event delivery parameter settings. Click Continue to return to the Edit Event Adapter page, and then click Back to return to the Final Configuration and Testing page.

   The Edit Event Adapter page allows you to define event parameters and configure the information that will be logged for the connection factory. Select one of the following settings for the log:

   – Log errors and audit messages

   – Log warnings, errors, and audit messages

   – Log informational, warning, error, and audit messages

   – Log all messages

Note: For maximum tracing, select Log all Messages. This is the recommended setting to use when you are collecting debugging information for BEA support.

The table that follows describes the type of information that each logging message contains.

Table 3-5 Logging message categories

This type of message	Contains
Audit	Extremely important information related to the business processing performed by an adapter.
Error	Information about an error that has occurred in the adapter, which may affect system stability.
Warning	Information about a suspicious situation that has occurred. Although this is not an error, it could have an impact on adapter operation.
Information	Information about normal adapter operations.

Testing Services

This information applies to "Step 8A, Test an Application View's Services" in *Using the Application Integration Design Console*, at the following URL:

`http://edocs.bea.com/wli/docs81/aiuser/index.html`

The purpose of testing an application view service is to evaluate whether that service interacts properly with the target FIX system. When you test a service, you supply any inputs required to start the service. For the Adapter for FIX, the input is in the form of a valid XML string that acts as input for the service.

Note: You can test an application view only if it is deployed and only if it contains at least one event or service.

To test a service:

1. In the Application View Administration page, click the Test link beside the service to be tested.

   The Test Services page appears.

2. In the Test Service window, copy the appropriate XML strings for the service. For example:

```
<?xml version="1.0" encoding="UTF-8"
?><FIXMLMessage><Header><Sender><CompID>
............56
.........</CompID>
      </Sender><Target><CompID>
............B2B
.........</CompID>
      </Target><SendingTime>
.........20031119-22:32:58
......</SendingTime>
   </Header><ApplicationMessage><Order><ClOrdID>
............BUY1069281178615
.........</ClOrdID><HandInst Value="1"/><Instrument><Symbol>
..............C
............</Symbol>
         </Instrument><Side Value="1"/><OrderQuantity><OrderQty>
..............100
............</OrderQty>
         </OrderQuantity><OrderType><MarketOrder Value="1"/>
         </OrderType><OrderDuration><TimeInForce Value="0"/>
         </OrderDuration><Currency
Value="USD"/></Order></ApplicationMessage></FIXMLMessage>
```

3. Click Test.

   The results appear in the Test Results window.

Testing a Buyside Event Using a Sellside Event

This information applies to "Step 8B, Test an Application View's Events" in *Using the Application Integration Design Console*, at the following URL:

http://edocs.bea.com/wli/docs81/aiuser/index.html

The purpose of testing an application view event is to make sure that the adapter correctly handles events generated by FIX. When you test an event, you can trigger the event using another event or manually, using a FIX client. You can test a buyside event by triggering a sellside event. In order to test a buyside event with a sellside event, you must configure the sellside event to send FIX messages to the host and port on which the buyside event is listening.

Note: You can test an application view only if it is deployed and only if it contains at least one event or service.

To test a buyside event using a sellside event:

1. In the Application View Administration page, click the Test link beside the buyside event to be tested.

   The Test Events page appears.

2. In the Time field, enter a reasonable period of time to wait, specified in milliseconds, before the test times out (One second = 1000 milliseconds. One minute = 60,000 milliseconds.).

3. Place a FIXML file in the OutboundMessageRepository of the Sellside event. You can do this either with an adapter service or you can place this file in the repository manually.

4. Click Test.

   The event is executed.

   – If the test succeeds, the Test Result page appears, showing the FIXML.

   – If the test fails, the Test Result page displays only a Timed Out message.

Testing a Buyside Event Using a FIX Client

This information applies to "Step 8B, Test an Application View's Events" in *Using the Application Integration Design Console*, at the following URL:

`http://edocs.bea.com/wli/docs81/aiuser/index.html`

The purpose of testing an application view event is to make sure that the adapter correctly handles events generated by FIX. When you test an event, you can trigger the event using a service or manually. You can test a buyside event by causing a FIX client to send a FIX message to the buyside event. In order to do this, you must configure the FIX client to send FIX messages to the host and port on which the buyside event is listening.

Note: You can test an application view only if it is deployed and only if it contains at least one event or service.

To test a buyside event using a FIX client:

1. In the Application View Administration page, click the Test link beside the buyside event to be tested.

   The Test Events page appears.

2. In the Time field, enter a reasonable period of time to wait, specified in milliseconds, before the test times out (One second = 1000 milliseconds. One minute = 60,000 milliseconds.).
3. Click Test.
4. Execute a FIX client program to send a FIX message to the host and port on which the buyside event is listening.

   The event is executed.

   – If the test succeeds, the Test Result page appears, showing the FIXML.

   – If the test fails, the Test Result page displays only a Timed Out message.

Testing a Sellside Event Using a Buyside Event

This information applies to "Step 8B, Test an Application View's Events" in *Using the Application Integration Design Console*, at the following URL:

`http://edocs.bea.com/wli/docs81/aiuser/index.html`

The purpose of testing an application view event is to make sure that the adapter correctly handles events generated by FIX. You can test a sellside event by triggering a buyside event. In order to use a buyside event to trigger a sellside event, you must configure the sellside event to send FIX messages to the buyside event's host and port.

Note: You can test an application view only if it is deployed and only if it contains at least one event or service.

To test a sellside event using a buyside event:

1. In the Application View Administration page, click the Test link beside the sellside event to be tested.

   The Test Events page appears.
2. In the Time field, enter a reasonable period of time to wait, specified in milliseconds, before the test times out (One second = 1000 milliseconds. One minute = 60,000 milliseconds.).
3. Place a FIXML file in the OutboundMessageRepository of the buyside event. You can do this either with an adapter service or you can place this file in the repository manually.
4. Click Test.

   The event is executed.

– If the test succeeds, the Test Result page appears, showing the FIXML.

– If the test fails, the Test Result page displays only a Timed Out message.

Testing a Sellside Event Using a FIX Server

1 2 3 4 5 6 7 **8** 9

This information applies to "Step 8B, Test an Application View's Events" in *Using the Application Integration Design Console*, at the following URL:

`http://edocs.bea.com/wli/docs81/aiuser/index.html`

The purpose of testing an application view event is to make sure that the adapter correctly handles events generated by FIX. You can test that a sellside event properly sends messages by causing it to send those messages to a FIX server. You can then examine those messages.You must configure the sellside event to send FIX messages to the host and port on which the FIX server is listening.

Note: You can test an application view only if it is deployed and only if it contains at least one event or service.

To test a sellside event using a FIX server:

1. Execute a FIX server program, and make sure it is listening for FIX messages.
1. In the Application View Administration page, click the Test link beside the sellside event to be tested.

   The Test Events page appears.

   Note: Make sure the sellside event is configured to send FIX messages to the host and port on which the FIX server is listening.
2. Click Test.
3. Place a FIXML file in the OutboundMessageRepository of the sellside event. You can do this either with an adapter service or you can place this file in the repository manually.

   The event sends the FIX message to the FIX server.
4. Check the FIX server program to make sure the FIX message was received.

Supported Messages

This section describes supported messages.

Message Categories, Types, and Descriptions

The packaging of FIX application messages in the `manifest.xml` are according the following categories:

Table A-1 Message Categories, Types, and Descriptions

Category	Message Type	Message Descriptions
Indication	Advertisements	Advertisement messages are used to announce completed transactions. The advertisement message can be transmitted in various transaction types; NEW, CANCEL and REPLACE.
	Indication of Interest (IOI)	Indication of interest messages are used to market merchandise, which the broker is buying or selling in either a proprietary or agency capacity. The indications can be time bound with a specific expiration value. Indications are distributed with the understanding that other firms may react to the message first and that the merchandise may no longer be available due to prior trade. Indication messages can be transmitted in various transaction types; NEW, CANCEL, and REPLACE.

Table A-1 Message Categories, Types, and Descriptions

Category	Message Type	Message Descriptions
Event Communication	News	The news message is a general free format message between the broker and institution. The message contains flags to identify the news item's urgency and to allow sorting by subject company (symbol). The News message can be originated at either the broker or institution side.
	Email	The email message is similar to the format and purpose of the News message, however, it is intended for private use between two parties.
Quotation	Quote Request	In some markets it is the practice to request quotes from brokers prior to placement of an order. The quote request message is used for this purpose. This message is commonly referred to as a Request For Quote (RFQ). Quotes can be requested on specific securities or forex rates. The quote request message can be used to request quotes on single products or multiple products. Securities quotes can be requested as either market quotes or for a specific quantity and side. If OrderQty and Side are absent, a market-style quote (bid x offer, size x size) will be returned.
	Quote Request Reject	The Quote Request Reject message is used to reject Quote Request messages for all quoting models.
	RFQ Request	In tradable and restricted tradable quoting markets, Quote Requests are issued by counterparties interested in ascertaining the market for an instrument. Quote Requests are then distributed by the market to liquidity providers who make markets in the instrument. The RFQ Request is used by liquidity providers to indicate to the market for which instruments they are interested in receiving Quote Requests. It can be used to register interest in receiving quote requests for a single instrument or for multiple instruments.

Table A-1 Message Categories, Types, and Descriptions

Category	Message Type	Message Descriptions
	Quote	The quote message is used as the response to a Quote Request message in both indicative, tradable, and restricted tradable quoting markets. In tradable and restricted tradable quoting models, the market maker sends quotes into a market as opposed to sending quotes directly to a counterparty. The quote message can be used to send unsolicited quotes in both indicative, tradable, and restricted tradable quoting markets. The quote message contains a quote for a single product.
	Quote Cancel	The Quote Cancel message is used by an originator of quotes to cancel quotes. The Quote Cancel message supports cancellation of: • All quotes • Quotes for a specific symbol or security ID • All quotes for a security type • All quotes for an underlying Canceling a Quote is accomplished by indicating the type of cancellation in the QuoteCancelType field.
	Quote Status Request	The quote status request message is used for the following purposes in markets that employ tradable or restricted tradable quotes: • For the issuer of a quote in a market to query the status of that quote (using the QuoteID to specify the target quote) • To subscribe and unsubscribe for Quote Status Report messages for one or more securities
	Quote Status Report	The quote status report message is used as: • the response to a Quote Status Request message • the response to a Quote Cancel message

Table A-1 Message Categories, Types, and Descriptions

Category	Message Type	Message Descriptions
	Mass Quote	The Mass Quote message can contain quotes for multiple securities to support applications that allow for the mass quoting of an option series. Two levels of repeating groups have been provided to minimize the amount of data required to submit a set of quotes for a class of options (for example, all option series for WCOM).
	Mass Quote Acknowledgement	Mass Quote Acknowledgement is used as the application level response to a Mass Quote message.
Market Data	Market Data Request	Some systems allow the transmission of real-time quote, order, trade and/or other price information on a subscription basis. A Market Data Request is a general request for market data on specific securities or forex quotes. A successful Market Data Request returns one or more Market Data messages containing one or more Market Data Entries. Each Market Data Entry is a Bid, an Offer, a Trade associated with a security, the opening, closing, or settlement price of a security, the buyer or seller imbalance for a security, the value of an index, or the trading session high price, low price, or volume weighted average price (VWAP).
	Market Data Snapshot/Full Refresh	Market Data messages can take two forms. The first Market Data message format used for a Snapshot, or a Snapshot + Updates.
	Market Data Incremental Refresh	The Market Data message for incremental updates may contain any combination of new, changed, or deleted Market Data Entries, for any combination of instruments, with any combination of trades, imbalances, quotes, index values, open, close, settlement, high, low, and VWAP prices, so long as the maximum FIX message size is not exceeded. All of these types of Market Data Entries can be changed and deleted.

Table A-1 Message Categories, Types, and Descriptions

Category	Message Type	Message Descriptions
	Market Data Request Reject	The Market Data Request Reject is used when the broker cannot honor the Market Data Request, due to business or technical reasons. Brokers may choose to limit various parameters, such as the size of requests, whether just the top of book or the entire book may be displayed, and whether Full or Incremental updates must be used.
Security and Trading Definition/Status	Security Definition Request	The Security Definition Request message is used to request a specific Security to be traded with the second party. The request security can be defined as a multileg security made up of one or more instrument legs.
	Security Definition	The Security Definition message is used for the following: • Accept the security defined in a *Security Definition* message • Accept the security defined in a *Security Definition* message with changes to the definition and/or identity of the security • Reject the security requested in a *Security Definition* message
	Security Type Request	The Security Type Request message is used to return a list of security types available from a counterparty or market.
	Security Types	The Security Type message is used to return a list of security types available from a counterparty or market.
	Security List Request	The Security List Request message is used to return a list of securities from the counterparty that match criteria provided on the request.
	Security List	The Security List message is used to return a list of securities that matches the criteria specified in a Security List Request.
	Derivative Security List Request	The Derivative Security List Request message is used to return a list of securities from the counterparty that match criteria provided on the request.

Table A-1 Message Categories, Types, and Descriptions

Category	Message Type	Message Descriptions
	Derivative Security List	The Derivative Security List message is used to return a list of securities that matches the criteria specified in a Derivative Security List Request.
	Status Security Request	The Security Status Request message provides for the ability to request the status of a security. One or more Security Status messages are returned as a result of a Security Status Request message.
	Security Status	The Security Status message provides for the ability to report changes in status to a security. The Security Status message contains fields to indicate trading status, corporate actions, financial status of the company. The Security Status message is used by one trading entity (for instance an exchange) to report changes in the state of a security.
	Trading Session Status Request	The Trading Session Status Request is used to request information on the status of a market. With the move to multiple sessions occurring for a given trading party (morning and evening sessions for instance) there is a need to be able to provide information on what product is trading on what market.
	Trading Session Status	The Trading Session Status provides information on the status of a market. With the move to multiple sessions occurring for a given trading party (morning and evening sessions for instance) there is a need to be able to provide information on what product is trading on what market.
Single/General Order Handling	New Order	The new order message type is used by institutions wishing to electronically submit securities and forex orders to a broker for execution. The New Order message type may also be used by institutions or retail intermediaries wishing to electronically submit Collective Investment Vehicle (CIV) orders to a broker or fund manager for execution.

Table A-1 Message Categories, Types, and Descriptions

Category	Message Type	Message Descriptions
	Execution Report	The execution report message is used to: • Confirm the receipt of an order • Confirm changes to an existing order (that is, accept cancel and replace requests) • Relay order status information • Relay fill information on working orders • Relay fill information on tradable or restricted tradable quotes • Reject orders • Report post-trade fees calculations associated with a trade
	Don't Know Trade (DKT)	The Don't Know Trade (DK) message notifies a trading partner that an electronically received execution has been rejected. This message can be thought of as an execution reject message.
	Order/Cancel/Replace Request	The order cancel/replace request is used to change the parameters of an existing order.
	Order Cancel Request	The order cancel request message requests the cancellation of all of the remaining quantity of an existing order.
	Order Cancel Reject	The order cancel reject message is issued by the broker upon receipt of a cancel request or cancel/replace request message, which cannot be honored. Requests to change price or decrease quantity are executed only when an outstanding quantity exists. Filled orders cannot be changed (for example, quantity reduced or price change. However, the broker/sellside may support increasing the order quantity on a currently filled order).
	Order Status Request	The order status request message is used by the institution to generate an order status message back from the broker.

Table A-1 Message Categories, Types, and Descriptions

Category	Message Type	Message Descriptions
	Order Mass Cancel Request	The order mass cancel request message requests the cancellation of all of the remaining quantity of a group of orders matching criteria specified within the request. NOTE: This message can only be used to cancel messages (reduce the full quantity).
	Order Mass Cancel Report	The Order Mass Cancel Report is used to acknowledge an Order Mass Cancel Request. Note that each affected order that is canceled is acknowledged with a separate Execution Report or Order Cancel Reject message.
	Order Mass Status Request	The order mass status request message requests the status for orders matching criteria specified within the request.
Cross Orders	New Order - Cross	Used to submit a cross order into a market. The cross order contains two order sides (a buy and a sell).
	Cross Order Cancel/Replace Request	Used to modify a cross order previously submitted using the New Order - Cross message. See Order Cancel Replace Request for details concerning message usage. Refer to the Order Cancel Replace Request (a.k.a. Order Modification Request) message for restrictions on what fields can be changed during a cancel replace.
	Cross Order Cancel Request	Used to fully cancel the remaining open quantity of a cross order.

Table A-1 Message Categories, Types, and Descriptions

Category	Message Type	Message Descriptions
Multi Leg Orders	New Order - Multileg	The New Order - Multileg is provided to submit orders for securities that are made up of multiple securities, known as legs. Swaps, option strategies, futures spreads, are a few examples of multileg securities. A multileg security is made up of multiple securities that are traded atomically. This requirement that all legs be traded in the quantities that they make up the multlileg security is the important distinction between a multileg order and a list order. Two generalized approaches to trading multileg securities are supported by FIX. The first approach involves a market maintaining multileg securities as separate products for which markets can be created. This "product approach" is often used in electronic trading systems. The second approach is to trade the multileg security as a group of separate securities – as is commonly done today in open outcry markets.
	Multi-leg Order Cancel/Replace Request	Used to modify a multileg order previously submitted using the New Order - Multileg message. See Order Cancel Replace Request for details concerning message usage.
List Program Basket Trading	Bid Request	The BidRequest Message can be used in one of two ways depending on which market conventions are being followed. In the "Non disclosed" convention (for example, US/European model) the BidRequest message can be used to request a bid based on the sector, country, index and liquidity information contained within the message itself. In the "Non disclosed" convention the entry repeating group is used to define liquidity of the program. In the "Disclosed" convention (for example, Japanese model) the BidRequest message can be used to request bids based on the ListOrderDetail messages sent in advance of BidRequest message. In the "Disclosed" convention the list repeating group is used to define which ListOrderDetail messages a bid is being sort for and the directions of the required bids.

Table A-1 Message Categories, Types, and Descriptions

Category	Message Type	Message Descriptions
	Bid Response	The Bid Response message can be used in one of two ways depending on which market conventions are being followed: • In the "Non disclosed" convention the Bid Response message can be used to supply a bid based on the sector, country, index and liquidity information contained within the corresponding bid request message • In the "Disclosed" convention the Bid Response message can be used to supply bids based on the List Order Detail messages sent in advance of the corresponding Bid Request message.
	New Order - List	The NewOrderList Message can be used in one of two ways depending on which market conventions are being followed: • In the "Non disclosed" convention the New Order - List message is sent after the bidding process has been completed, by telephone or electronically. The New Order - List message enumerates the stocks, quantities, direction for the trade and may contain pre-allocation information. This message may also be used as the first message for the transmission of a program trade where the bidding process has been done by means other than FIX. In this scenario the messages may either be used as a staging process, in which case the broker will start execution once either a ListExecute is received or for immediate execution, in which case the orders will be executed on receipt. • In the "Disclosed" convention the New Order - List message is sent before the bidding process is started, by telephone or electronically. The New Order - List message enumerates the stocks and quantities from the bidding process, and may contain pre-allocation information. The direction of the trade is disclosed after the bidding process is completed.

Table A-1 Message Categories, Types, and Descriptions

Category	Message Type	Message Descriptions
	List Strike Price	The strike price message is used to exchange strike price information for principal trades. It can also be used to exchange reference prices for agency trades.
	List Status	The list status message is issued as the response to a List Status Request message sent in an unsolicited fashion by the sell-side. It indicates the current state of the orders within the list as they exist at the broker's site. Orders within the list are statused at the summary level. Individual executions are not reported, rather, the current state of the order is reported.
	List Execute	The list execute message type is used by institutions to instruct the broker to begin execution of a previously submitted list. This message may or may not be used, as it may be mirroring a phone conversation.
	List Cancel Request	The list cancel request message type is used by institutions wishing to cancel previously submitted lists either before or during execution.
	List Status Request	The list status request message type is used by institutions to instruct the broker to generate status messages for a list.

Table A-1 Message Categories, Types, and Descriptions

Category	Message Type	Message Descriptions
Allocation and Ready To Book	Allocation	The Allocation message provides the ability to specify how an order or set of orders should be subdivided amongst one or more accounts. It can also be used as a confirmation message through which third parties can communicate execution and settlement details between trading partners. In addition, the allocation message can be sent by the broker to communicate fees and other details that can only be computed once the sub-account breakdowns are known. Note the response to the Allocation message is the AllocationACK message. The Allocation message can also be sent by the buyside firm after execution to indicate to the sellside firm that one or a combined (aggregated) set of orders are "Ready-To-Book" without specifying individual account breakdowns. This can be used to trigger post-trade allocation, matching, and settlement processing via other channels (for example, post-trade industry utilities).
	Allocation ACK	The Allocation ACK message is used to acknowledge the receipt and status of an Allocation message. It is possible that multiple Allocation ACK messages can be generated for a single allocation to detail the receipt and then the acceptance or rejection of the Allocation message.
Settlement Instructions	Settlement Instructions	The Settlement Instructions message provides the broker's, the institution's, or the intermediary's instructions for trade settlement. The SettlInstSource field indicates if the settlement instructions are the broker's, the institution's, or the intermediary's. This message has been designed so that it can be sent from the broker to the institution, from the institution to the broker, or from either to an independent "standing instructions" database or matching system or, for CIV, from an intermediary to a fund manager.

Table A-1 Message Categories, Types, and Descriptions

Category	Message Type	Message Descriptions
Trade Capture Reporting	Trade Capture Report Request	The Trade Capture Report can be used to: • Request one or more trade capture reports based upon selection criteria provided on the trade capture report request • Subscribe for trade capture reports based upon selection criteria provided on the trade capture report request
	Trade Capture Report	The Trade Capture Report message can be: • Used to report trades between counterparties • Can be sent unsolicited between counterparties • Sent as a reply to a Trade Capture Report Request • Can be used to report unmatched and matched trades
Registration Instructions	Registration Instructions	The Registration Instructions message type may be used by institutions or retail intermediaries wishing to electronically submit registration information to a broker or fund manager (for CIV) for an order or for an allocation. A Registration Instructions message can be submitted as new, cancel or replace. The RegistTransType field indicates the purpose of the message.
	Registration Instructions Response	The Registration Instructions Response message type may be used by broker or fund manager (for CIV) in response to a Registration Instructions message submitted by an institution or retail intermediary for an order or for an allocation.

Index