



FIX for
BizTalk
Server
Whitepaper

An introduction

Michael Lipscombe, Synthesis Consulting, Inc

FIX for BizTalk (The basics)

FIX is a messaging standard and protocol that is used by the finance industry for real-time electronic exchange of securities transmissions. It was developed by the industry to facilitate the automated trading of financial instruments and is used around the world by banks, brokers, exchanges and order management and trading systems. The site fixprotocol.org has some good information to get you started.

To the uninitiated a FIX message will look a little strange but it won't take you too long to get your head around it. Below is an example of what an Order (NewOrderSingle) might look like:

```
1: 8=FIX.4.2 9=118 35=D 34=23 49=OMS 56=NSDQ 52=20100721-03:16:21.679 40=1 55=MSFT 54=1
60=1999-05-31T13:20:00.000-05:00 38=100 21=1 10=169
```

The numbers are referred to as Tags. Tag 8 at the start is the version which in this case is **FIX 4.2**. Tag 35 denotes the message type, **D** is NewOrderSingle. This is a BUY order (54=1) for 100 shares (38=100) of Microsoft (55=MSFT) at a market price (40=1). Tag 49 is SenderCompID and is a string that identifies the Sender of the message to the counter party or recipient. 56 is TargetCompID which is used by the counterparties FIX engine to verify the message. These values are important when setting up BizTalk which I'll cover in a later article.

To learn more about these tags and the messages that FIX supports there's a handy app online: [fixmate](http://fixmate.com).

FIXML

The FIX protocol guys have also come up with XML versions of FIX messages which is called **FIXML**. The NewOrderSingle above in FIXML would look like this:

```
1: <Order OrdTyp='1' Sym='MSFT' Side='1' TxnTm='1999-05-31T13:20:00.000-05:00'
2:     Qty='100' ClOrdID='' HandlInst='1'>
3:     <Hdr SndgTm='1999-05-31T13:20:00.000-05:00' SeqNum='23' SID='OMS' TID='NSDQ' />
4: </Order>
```

Nothing too scary, the emphasis is on small efficient messages as speed is the name of the game.

FIX Engines

A FIX engine is the application or service that enables the transmission of these messages between parties. The FIX engines connect over using a standard TCP/IP connection, they manage the session level communications (authentication, validation, heartbeats, session recovery) and the application layer (orders, executions etc).

FPL defines a FIX engine as:

"A FIX engine is a piece of software that manages a network connection, creates and parses outgoing and incoming messages, respectively, and recovers if something goes wrong. A FIX engine manages the session and application layers and is the single piece of software you need in order to FIX-enable trading or order management systems. In the context of a trading system your FIX engine is the interface to the outside world, which, together with a network, connects you to outside world and allows you to trade and exchange related information in a standard fashion. Thus, to FIX-enable an application refers to the integration of a FIX engine and connection to a routing network."

BizTalk

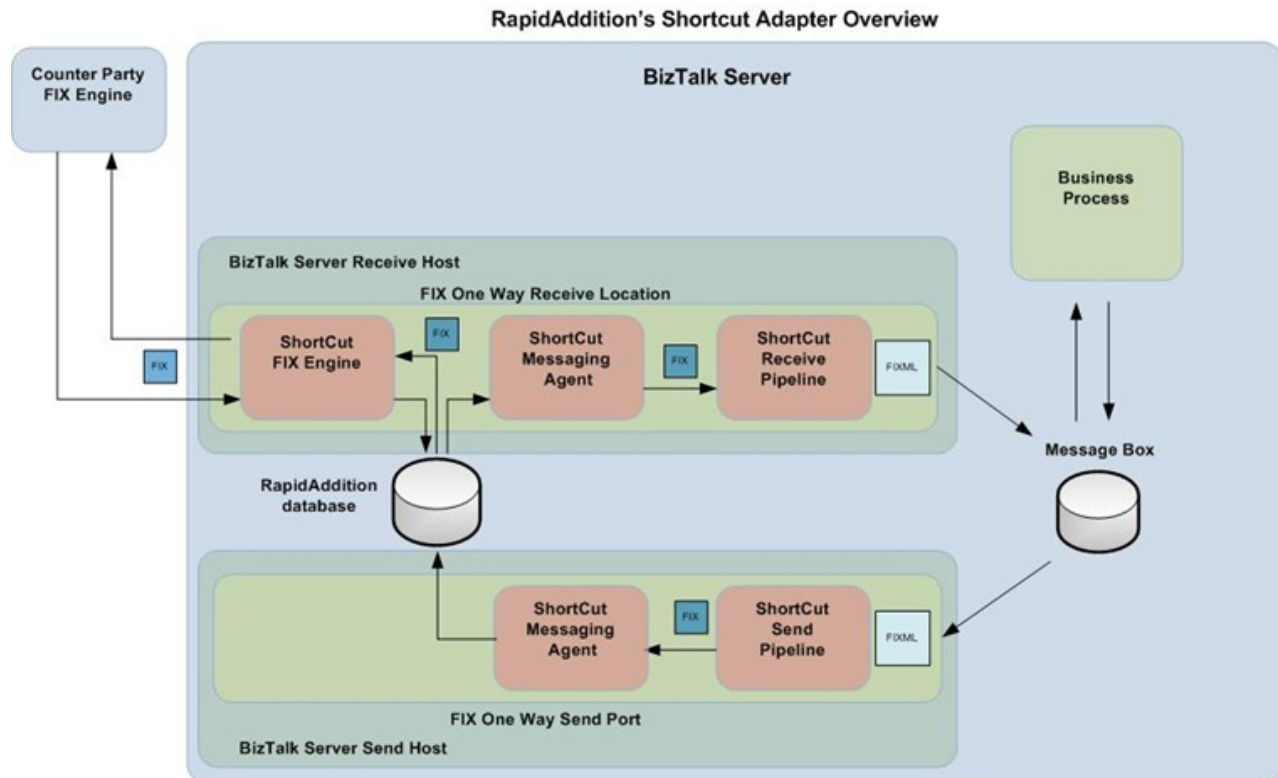
The folks at [RapidAddition](http://RapidAddition.com) have been building FIX engines since FIX was running around in diapers. They've put together a very nice FIX BizTalk Adapter that FIX-enables your BizTalk Server. Your first thought is probably "I thought BizTalk was slow and FIX was about speed", which just isn't true. I have worked on FIX-enabled BizTalk systems manage hundreds of executions a second. Whilst there are limitations it all comes down to how you configure BizTalk and design and build the application. A lot of that is contingent on what your application has to do. For example just routing orders will be super quick but if you need to crack open the message and perform validations; transformations etc then it will come at some cost.

RapidAddition's product is called [ShortCut](#) and is super easy to get started with. ShortCut will basically turn your Receive Locations into FIX engines. ShortCut will manage all the FIX session stuff and you just need to worry about routing or orchestrating application messages through your FIX hub. ShortCut comes with all the FIXML schemas in DLL's for you to reference in your projects. Building applications is pretty easy and you don't need to be a FIX weapon to do this as I hope to show in posts I'll put together over the next few weeks.

Overview

First we should start with an overview of the ShortCut FIX Adapter, the components that get installed and give a bit more detail on what's happening.

The diagram below gives an overview of the ShortCut adapter.



BizTalk only ever sees the FIXML version of the messages. The conversion to and from FIX takes place in the pipelines that come with ShortCut.

The receive location not only receives FIX messages but it also sends them out. The send port just persists the converted messages into the RapidAddition database and the engine reads them and sends it over the wire. FIX is always two way communications, even if there is only one party sending application messages. Most if not all admin messages that make up the session layer are acknowledged and there are heartbeats back and forth.

A BizTalk send port only activates when a message is picked published for it. It wouldn't be possible to have the send port be responsible for the outbound admin messages without BizTalk getting involved. Receive Ports on the other hand are always executing or polling. This then becomes the logical place to manage the session layer so it must also send out the application messages as they go over the same tcp/ip connection. Architecting the adapter this way enables the FIX engine to deal with all the FIX session layer and leaves BizTalk to only worry about the application messages.

From an Orchestration or send ports perspective it's business as usual. They can route and send messages the same as using any other adapter. Due to some requirements of the FIX protocol the adapter has to be architected this way to save the BizTalk developer having to deal with the session layer, which is a good thing. A normal FIX session might have 6 or so different admin messages zipping back and forth, if you're only interested in sending Orders then that's all your application has to worry about.

Something to keep in mind though is that you'll have to make sure the send and receive ports are running in the same host. The diagram above is actually a incorrect, but it keeps it simple for the initial explanation.

Another important FIX concept to know is that one of the parties is designated as the initiator of the session. That simply means one party always initiates the connection based on a predetermined schedule and the other is only ever listening. You configure this in the receive location which will be covered in another post.

Components

Database

As you've probably figured out ShortCut needs it own database. This is installed for you through a wizard which we'll cover later. Most of the time this will be put on the SQL server that's hosting the BizTalk databases. If your processing a large number of messages then it might be worth putting this on a separate server. You don't want BizTalk and ShortCut to fight for resources. Keep in mind that I've managed systems that process over 800,000 messages a day and it wasn't an issue so much as an area of potential improvement if the budget was available.

For DR purposes I would consider adding the RapidAddition database to the BizTalk log shipping jobs. During a failover you'll want the transactional state of this database to go with the whole BizTalk system. When FIX engines reconnect during the day they resync themselves based on the last processed sequence number which is stored in the database. Reconnecting from the DR site with a sequence number of 1 will cause all of the days messages to be resent which could be a nightmare.

Adapter

On the actual adapter side it's all just a bunch of assemblies nothing all that different from what you would expect. Some are BizTalk assemblies (schemas, promoted property schemas) that you'll reference in your projects and the rest are standard .NET dll's.

They also give you a handy FIX test harness for testing out your application without connecting to counterparties. It can do things like auto fill orders, reject all messages and even replay FIX logs in case you have some production logs you want to test with.

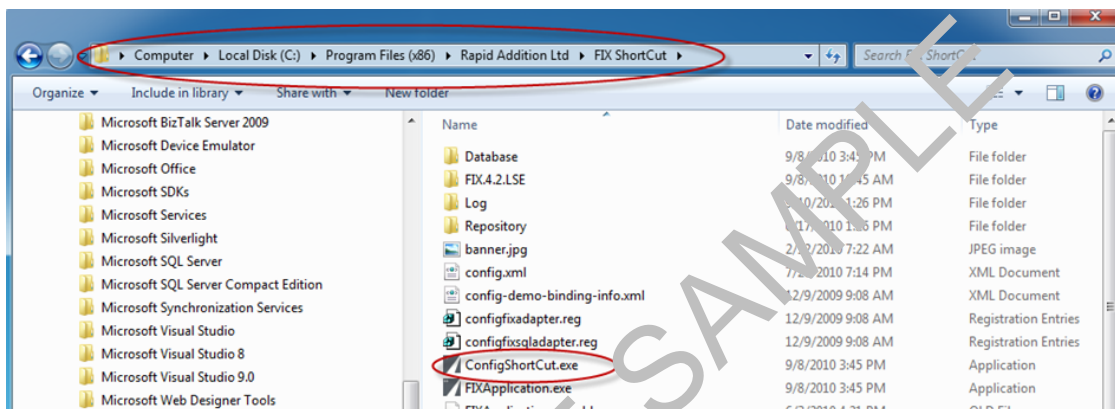
Installing and Configuring the FIX Adapter for BizTalk

This section will go through installing and configuring the RapidAddition's FIX adapter for BizTalk. The screenshots are from BizTalk Server 2009 however it's the same process for 2010 or even 2006.

Installing

Installing the Adapter is pretty easy. You'll get an MSI that will install the DLL's on to the BizTalk server. Shortcut does not support host clustering although you can install the FIX host on multiple machines so that you could quickly bring the session online if you lost a server. Each FIX session can only run in a single host and you can have multiple sessions per host.

Once you run through the installation you'll need to configure it. There's a configuration app in the program directory created during the installation.



The config app has 4 functions to run through. The first will install the FIX Schema and Pipeline assemblies on all servers in the BizTalk group and BTSDeploy's them. The second option will install the adapter into the registry and