

The logo for hatstand, featuring the word "hatstand" in a lowercase, serif font. The letters are a light green color, and the "h" and "a" are connected. The background behind the text is a light green gradient.

Specialists in Financial Technology

White Paper

Exchange Connectivity

**MAKE A SERIOUS DIFFERENCE TO  
YOUR PERFORMANCE**

## Table of Contents

Exchange Connectivity Technology .....	2
Building Exchange Links.....	3
Project Management .....	4
Analysis and Design.....	5
Design and Implementation .....	9
Testing.....	11
Documenting a Market Adaptor.....	13
Summary.....	13

**This document was written by Adam Bennett – Managing Partner .**

**Hatstand Limited  
32-38 Leaman Street  
London  
E1 8EW  
Tel +44 (0) 20 7173 6280  
Fax +44 (0) 20 7173 6284**

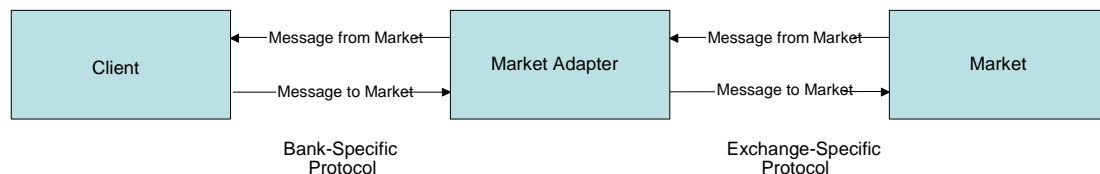
**E-mail [adam-bennett@hatstand-ltd.com](mailto:adam-bennett@hatstand-ltd.com)  
Web [www.hatstand-ltd.com](http://www.hatstand-ltd.com)**

## Exchange Connectivity Technology

Financial services organisations are under continued pressure to meet customer demands for new services whilst maintaining clear and auditable compliance with regulation that continues to evolve. IT directors are expected to consistently answer the challenge of maintaining a competitive technological edge in a cost-efficient way.

The 'traditional' approach to acquiring access to exchange trading hosts – via such trading platforms as GLTrade and Fidessa – has not always been seen to deliver the business benefits that matter in an increasingly volatile and competitive market. Being first to reach the market is in some cases, the only success criterion that counts. The need to deliver fast and scalable solutions has opened up an area of exciting technological growth: direct exchange links via bespoke market adaptor software. Market adaptors allow client systems to streamline functionality and performance by linking directly with the exchange, thereby maximising revenues whilst achieving economies of scale and lowering costs.

The market adaptor sits between the market connection and the client application, translating communications from the client and market so that communications from one can be received and understood by the other. The client communicates with the adaptor using one of a number of interfaces which may be in-house or common messaging standards such as FIX or ISO15022. The adaptor typically communicates with the market using the market's own protocol.



Market adaptors are widely considered to be the fastest vehicle for executing orders in the market, outperforming traditional trading platforms by a significant margin. The benefits of using direct exchange links are not just restricted to trading: market adaptors also provide some of the highest quality data currently available. Such data includes full market depth and every tick of data that the exchange publishes. This contrasts sharply with data vendors who aggregate data during busy periods rather than passing it on the client without delay. Market adaptor solutions help to reduce market data costs by avoiding costly exchange fees.

The performance of a market adapter is typically significantly faster than 3<sup>rd</sup>-party platforms such as Royal Blue, Fidessa and GL Trade.

## Building Exchange Links

Building exchange links is a complex task requiring specialist knowledge, skill and time that is not always available or effective to undertake in-house. When outsourcing the development of exchange connectivity what should your selection criteria be?

- Find a supplier with exchange connectivity solutions that provide access to global markets (including bonds, equities and derivatives). That provides expertise in all areas of the product lifecycle, from project feasibility and business case development, through analysis and implementation to testing and support. IT directors may choose to implement either part or the entire spectrum of functionality, including market execution, market data and research.
- Make experience high priority. Look for a consultancy who have worked with global financial institutions using their applied knowledge of the financial services market. That can draw on decades of investment banking experience, alongside proven technical, project management and QA expertise to deliver projects on time and to budget. This significantly reduces the problems associated with Greenfield projects where the risks and issues cannot be dealt with until they are encountered, by which time it often too late.
- Search for a supplier who can combine a successful quality approach with the latest development techniques and methodologies. The historical evolution of market adaptor development has wrought significant changes: the design pattern now in use makes far greater use of object orientation, library support, and the features of the C++ language.
- Make sure your supplier adds value by providing more than the minimum set of key deliverables, including a Proof of Concept document, a detailed project plan, analysis documentation (including a detailed specification), test pack (including outline for conformance testing), source code and operations guide

In this article we examine a cost effective and timely approach to market adaptor software development. Each chapter presents a key aspect of the development. Key features include, project management, analysis and design, implementation, testing and support.

## Project Management

Your project manager needs to understand the business and technology issues. Risks and project interdependencies need to be highlighted and discussed with the client prior to implementation, and are included in the risks and issues log which is created at the start of the project and tracked throughout the project lifecycle. Typical project issues and dependencies include:

- Access to relevant and up-to-date technical documentation from the exchange
- Client network configuration
- Configuration of hardware (such as development servers, testing servers etc)
- Sign-off of project phases and achievement of quality gates
- Selection of third-party software required
- Necessary third-party software acquired, installed and configured where necessary.
- Working within the various constraints imposed by the exchanges, such as conformance testing and developing functionality required by the client that the exchange does not provide.
- Differences occurring between development and production, such as the use of ISDN in development, but a higher capacity leased-line in production. Different bandwidths being available in the different environments, meaning that under high-load, timeouts and/or heartbeat issues become an issue in one environment but potentially not the other.
- Upgrades, mandatory and otherwise.

The amount of time taken to deliver a market adapter varies with each market and set of functional requirements and may take between 2 weeks and 3 months.

All aspects of project governance are managed including specialist project management areas such as change management and risk management.

## Analysis and Design

The business and/or technical analysts need to produce a detailed specification of the required market adaptor using data flow analysis and translation as part of the service offering. The specification will contain details of any data caches to be implemented, as well as how any required functionality that does not map directly onto that provided by the exchange is to be simulated where required by the client. The analysis should be documented using a mixture of tools such as Rational Rose, Word, Visio and Excel.

The client communicates with the adaptor using either a high speed internal interface specific to the organisation or a standard communications protocol such as FIX or ISO 15022. Ideally, the adaptor should implement as many interfaces as the adapter may be required to use so that any client system that uses one of the implemented interfaces as a communications protocol may communicate with it. Multiple interfaces may need to be implemented within the adaptor for a number of reasons.

The client may require the adaptor to fulfil specific functions using dedicated protocols (e.g. one protocol used for trading and one used for market data), or the adaptor may be required to connect to multiple hosts that use different protocols. Conversely, a separate adaptor may be implemented to handle each requirement if the resulting architecture is unnecessarily complex or requirements are varied. For example, a high-speed adaptor may required for execution, while a real-time adaptor may be required for market data. The market adaptor should be able to cope with both types of types of performance requirements. The bulk of the specification will contain details of the mapping between the interface used by the client system, and the interface used at the exchange.

The messages that the client can send to the market via the market adaptor allow the client to submit new orders, amend orders, cancel orders, submit quotes and request market data etc. The messages that the client can receive from the market via the market adaptor include executions, acknowledgements, rejections, prices etc.

As well as providing a translation facility, the market adaptor is also responsible for managing the connection and logon session with the market. Connection is the behaviour of establishing a connection with the market, and logon is the behaviour of authenticating the user over this connection.

You need to develop market adaptors that are capable of recovering their view of the client's status in the market. Recovering information about quotes, orders and trades that the client

system is interested in are included as some of the fundamental aspects of the adaptor's functionality.

The underlying market may or may not support all of the required functionality. The analysis process required to specify a market adaptor involves working out how to present that standard interface to the client for a particular market.

The process of specifying a market adaptor is an iterative one. For each of the messages that the Client intends to send to the adaptor and for each message that the Client expects to receive back, a mapping process has to occur. The analyst will attempt to match functionality in the interface against functionality in the market's interface. Where a match exists, a translation can occur and the message passed.

Market adaptors are event-driven, where the events include receipt of messages from the client, receipt of messages from the server, connections failing, timeouts occurring etc. They are also data-driven, with the relationship between different data items being an essential part of the analysis. No one tool that is currently in use by the market adaptors development group can document such a system well.

- UML sequence diagrams are useful for showing the relationship between messages between client and adaptor, and between adaptor and market.
- UML Class diagrams are useful for showing the contents of messages, the designs of caches, the relationships between those items etc.
- Tables are useful for showing how fields in a message can be populated, identifying precisely where the data to be used comes from and describing any translations or other rules required.

Although not the perfect tool for this job, Rational Rose is certainly adequate. It allows the production of the UML items mentioned, and allows both logical and physical diagrams to be produced (which is useful when optimisations are included). Storing tables within Rational Rose is, however, not ideal. However, even though it does not present tables well, it is felt best to keep them in Rational Rose so that the full analysis documentation is in one place.

An example market adaptor topology is illustrated in the figure below:

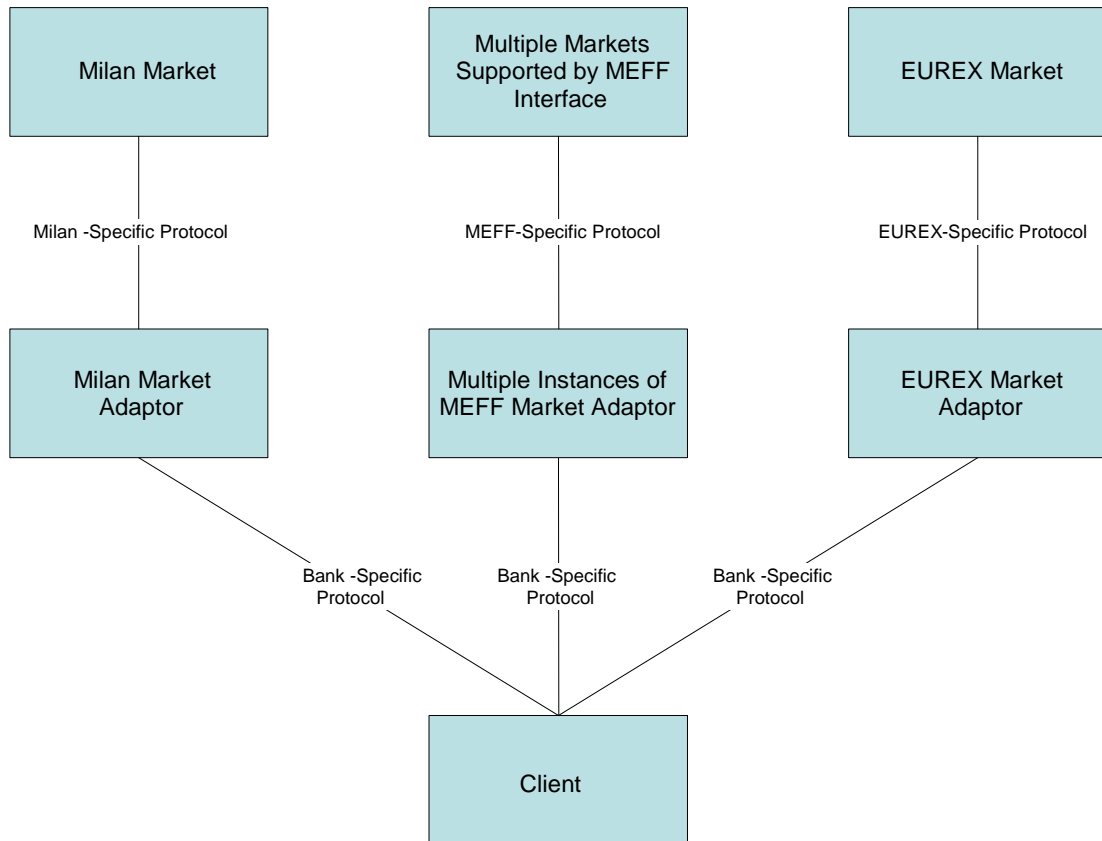
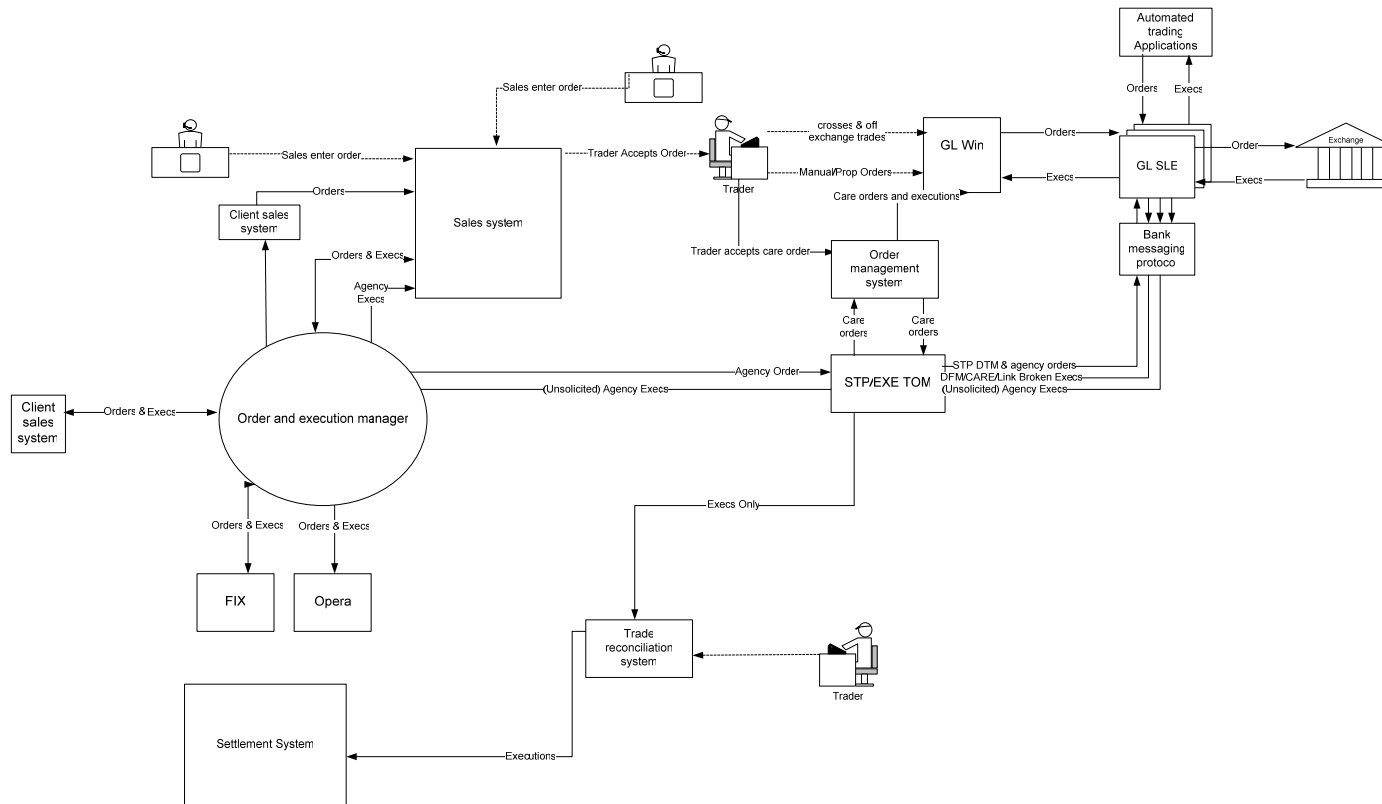


Figure 1

An example business message flow is shown below, incorporating both 3<sup>rd</sup> party and market adaptor message communications:



## Design and Implementation

The key requirements to bear in mind when designing a market adaptor are:

- The latency of messages passing through the market adaptor must be minimised, particularly for messages going from client to market
- The market adaptor must be capable of maintaining a high transaction rate
- Data quality must be preserved

Performance tuning will necessarily be carried out throughout the development of the market adaptor in order to achieve the above objectives.

A number of critical features need to be borne in mind during development,:

- Threading. Market adaptors are multi-threaded, and normal multi-threaded development rules should be followed regarding avoiding race conditions, deadlocks etc. The market adaptor follows any rules specified by the exchange regarding how to use threads when communicating with the market.
- Context Switching.
- Performance. Within the design constraints imposed ensure that the market adaptor meets the performance criteria set in terms of latency and transaction rate.
- Data Caching. You should follow the general rule-of-thumb that only as much data as will allow the adaptor to function correctly should be persisted so as not to impact performance. This depends on the functionality supported by the underlying exchange.
- Handling Different Network Characteristics. Different types of network connection exhibit difference characteristics. It is important to remember that development and production environments may use different networking solutions and this needs to be taken into account when designing / coding a market adaptor.
- Portability. Many C/C++ market adaptors are currently being developed to execute on Solaris. However, as more and more organisations migrate to Linux, market adaptors should be designed / developed with this in mind.

Your developers should have knowledge of developing on a range of platforms, from Solaris or Linux to Windows. They should make expert use of relevant tools, such as:

- Solaris 2.8
- Sun Forte C++ compiler
- Rational Purify
- Rational Quantify
- Rational Pure Coverage
- Flexelint
- Clearcase
- ACE libraries
- Boost libraries
- Doxygen
- Borland JBuilder
- Websphere
- Javadoc.

Development tasks include:

- Defining and constructing a market adaptor
- Using the basic functions of a market adaptor
- Managing a set of properties
- Using 'helper' classes for tasks such as logging and message queuing
- Implementing the relevant communications protocol
- Using the factory pattern for market adaptor instantiation

## Testing

Testing is a well-documented part of the project lifecycle, so this article is not going to repeat basic testing methods (for a good introduction to testing see "Software Testing Techniques" by Boris Beizer). Instead, this section describes aspects of testing specific to market adaptors.

You should seek to ensure that even before development starts, a valid connection to the market is in place. Even if a market simulator can be installed locally, it is useful to be able to connect to the market from day one so that the behaviour of both the real market and the connection can be observed throughout development and testing.

You should also ensure that a machine is available that has the same characteristics as the target platform. If the target platform is a multi-processor machine, ensure that development and testing are also performed on a multi-processor machine.

Multiple test harnesses with varying behaviour are also developed in order to exercise the market adaptor as much as possible. This is explained in further detail in the sections that follow.

### ***Quantitative Analysis***

Early in the project, you should develop a suitable harness for timing pieces of code, measuring how often C++ mutexes are locked/unlocked, measuring how often objects are constructed/destroyed etc.

### ***Functional/System Testing***

It is considered beneficial to develop several types of test harness or client:

- A test client that simulates the client system's behaviour as closely as possible, including linking in any libraries that the final production client uses
- A harness that allows one or more messages to be passed through the market adaptor under user control, with responses logged and/or validated and/or output to the console
- A harness that exercises the functionality of the market adaptor without user intervention. The functionality supported by different market adaptors varies depending on the underlying market, but a starting point for such a harness would be to instantiate the market adaptor, connect, logon, download listings, subscribe to updates, then as prices are received send orders, randomly cancel orders etc.

### ***Integration Testing***

During Integration Testing, ensure that the test client being used links in any libraries that the production Client is expected to use. This allows any conflicts to be detected prior to User Acceptance Testing.

### ***Performance Testing***

Measure the round-trip time of a message and its response through the market adaptor, attempting to exclude as much time spent outside the market adaptor as possible.

Note however, that developing a harness that simulates the market just enough to allow the performance test to be executed can involve a reasonable amount of effort. If timescales do not permit this to be done, another method will need to be identified. However, this remains the preferred method of performance testing.

### ***Static Analysis***

Use static analysis tools such as Flexelint to look for common problems within the market adaptor code. It is not necessary to change code to get rid of every warning raised by Flexelint, but do investigate the cause of every one before either changing the code or using a Flexelint flag to disable it.

If tools are available on other platforms, remember that the vast majority of market adaptor code will port very easily to other platforms. The vast majority market adaptor functionality is tested on both UNIX and Windows platforms, allowing it to be passed through a number of tools, including two C++ compilers (each with different warning capabilities), Purify, BoundsChecker and PC-Lint.

### ***Dynamic Analysis***

Use dynamic analysis tools such as Purify and Quantify to look for resource leaks, areas of code where further optimisation is required, areas of code that either get called more often than expected or not at all etc.

## Documenting a Market Adaptor

Ensure your development is properly documented. The following is a list of typical documents:

Document Name	Contents
Requirements	Specifies the high-level objectives of the market adaptor, including performance requirements, target platform (machine details, operating system, other software that it must not conflict with etc), functionality to support etc
Project plan including milestones	Shows what tasks needs to be done to meet the requirements
Analysis documentation, including sequence diagrams, class diagrams, translation tables	Detailed breakdown of what the system needs to do
Design	Shows how the system will implement the items listed in the analysis
Test plan	Documents how testing will be done
Test scripts	The information required to perform each test
Development-production transition checklist	A checklist that contains a breakdown of everything that needs to be done to ensure a smooth transition from the development environment into production
Operations guide	Instructions on how to configure / use the completed market adaptor

Code is thoroughly commented, using Doxygen where useful.

## Summary

As the structure of the financial services market continues to evolve, the drive to maintain a competitive edge demands constant innovation and streamlining of processes from IT heads. We have seen in the above article that market adaptors are a cost-effective vehicle for delivering crucial business benefits for organisations looking to implement market data and execution.

The successful development and implementation of market adaptors is dependant on a skilled and experienced team working to proven methodologies.

## About Hatstand

Hatstand has proven business and technical expertise that is utilised to consistently deliver our clients improved performance on time and to budget.

Hatstand has delivered market adaptor projects for the following exchanges:

- Amsterdam
- Brussels
- Chicago
- Copenhagen
- Frankfurt
- Helsinki
- Hong Kong
- Johannesburg
- Madrid
- Milan
- Madrid
- New York
- Osaka
- Oslo
- Paris
- Stockholm
- Virt-x

Hatstand's unique service offering is predicated on its ability to offer a range of skills and expertise covering the three dimensions of IT skills, IT functions and business knowledge. Hatstand's 3D-IT™ approach has proven to be most effective at plugging the gaps on any project, supplementing even the most complex requirements with quality resources at short notice and at low cost.

Business Functions / Environments:	IT Functions Covered:	IT Skills Available:
<ul style="list-style-type: none"> <li>• System integration</li> <li>• High Availability</li> <li>• High Performance</li> <li>• Client connectivity</li> <li>• Exchange links</li> <li>• Order routing</li> <li>• Trade order management</li> <li>• Vendor/exchange liaison</li> <li>• Trade order routing</li> <li>• FIX engines</li> <li>• Equities</li> <li>• Fixed Income</li> <li>• Derivatives</li> <li>• Credit Derivatives</li> <li>• Direct Market Access</li> </ul>	<ul style="list-style-type: none"> <li>• Business analysis</li> <li>• Coding</li> <li>• Design</li> <li>• Testing</li> <li>• Implementation</li> <li>• Handover</li> <li>• Support</li> <li>• Project Management</li> <li>• Performance review</li> <li>• Methodologies</li> <li>• Process design/review</li> </ul>	<ul style="list-style-type: none"> <li>• C++</li> <li>• Java</li> <li>• Unix</li> <li>• Multi-threading</li> <li>• High performance</li> <li>• High availability</li> <li>• TRIARCH-SSL</li> <li>• TIBCO-Rendezvous</li> <li>• MQ Series</li> <li>• JNI</li> <li>• Oracle</li> <li>• Sybase</li> <li>• SSH, SSL-Security</li> <li>• XML, Perl</li> <li>• TLP-IP</li> <li>• VBA</li> <li>• FIX</li> </ul>

The following benefits are available to the client that chooses Hatstand to deliver market adapter software:

- Outperform traditional trading platforms by a significant margin
- Receive accurate, consistently high-quality data
- Reduce costs associated with acquiring market data by having membership at the relevant exchange
- Receive full market depth and every tick of data that the exchange publishes
- Decades of investment banking experience at your disposal
- Utilise proven expertise in all areas of the product lifecycle
- Applied knowledge of the financial services market and decades of investment banking experience.