

## **SuperFeed Client Specification**

**Version 1.1**

**July 1, 2010**

## PREFACE

### DOCUMENT HISTORY

Document Version	Date	Change Description
1.0	03/19/2010	Initial distribution
1.0a	04/07/2010	Further Information URLs updated
1.1	07/01/2010	Added 4.1 Windows Tuning Parameters

### CONTACT INFORMATION

For technical support please contact the Global Service Desk:

- Telephone: +1 212 383 3640 (International)
- Telephone: 866 873 7422 (Toll free, US only)
- Email: [service.desk@nyx.com](mailto:service.desk@nyx.com)

### FURTHER INFORMATION

- SuperFeed product page: <http://www.nyxdata.com/superfeed/>
- Data Fabric Product Sheet: <http://www.nyxdata.com/productsheets/datafabric>
- MAMA Product Sheet: <http://www.nyxdata.com/productsheets/mama>
- MAMDA Product Sheet: <http://www.nyxdata.com/productsheets/mamda>
- MAMA and MAMDA developer documentation: <http://www.nyxdata.com/apis/>
- Capacity figures: <http://www.nyxdata.com/capacity/>
- Glossary: <http://www.nyxdata.com/glossary/>
- Web Issue Management System: <http://issues.wombatfs.com/> (login required)

## TABLE OF CONTENTS

<b>1</b>	<b>Product Overview .....</b>	<b>4</b>
1.1	Technology Overview .....	4
1.1.1	Platform Flexibility .....	5
1.1.2	Data Sourcing and Entitlements .....	7
1.2	APIs: MAMA and MAMDA .....	8
1.2.1	Entitlement Checking .....	9
1.3	Global Market Data Administration Team Services .....	10
1.4	Connectivity .....	10
1.5	market Coverage .....	10
<b>2</b>	<b>Platform Components.....</b>	<b>11</b>
2.1	Feed Handlers .....	11
2.2	SuperBook .....	11
2.2.1	Entry IDs.....	11
2.2.2	Entry ID Example .....	11
2.3	MamaCache.....	14
2.4	SuperFeed Feed Handler .....	15
2.5	Entitlement Server .....	15
2.6	MAMA and MAMDA APIs .....	15
2.6.1	Development Tips and Guidelines .....	15
<b>3</b>	<b>Support .....</b>	<b>17</b>
3.1	Web Issue Management System .....	17
3.2	Telephone .....	17
3.3	Email.....	17
3.4	Monitoring .....	18
3.5	Escalation .....	18
<b>4</b>	<b>System Requirements and Recommendations.....</b>	<b>19</b>
4.1	Windows Tuning Parameters.....	20
4.2	MAMA and MAMDA Requirements .....	20
4.3	Capacity .....	20

## 1 PRODUCT OVERVIEW

SuperFeed™ provides customers with broad access to data from major US, European and Asian markets in a single normalized format. This hosted, managed service provides the simplicity and cost efficiency of a consolidated feed with the high performance capabilities of a direct feed. It was developed to meet client demand for a single-vendor solution that provides cost-efficient access to multiple markets via ultra low latency connections (via Secure Financial Transaction Infrastructure® (SFTI®) network or co-location). SuperFeed provides both normalized and value-added market data over a feature-rich, unified API interface, the Middleware Agnostic Messaging API (MAMA), ensuring both backwards compatibility with in-house ticker plants and a future-proof interface designed to support changes and new content.

For the first time, market participants can now access the high performance and expansive market reach of NYSE Technologies' market data software with the economics and convenience of a hosted, managed service. This allows clients to focus on their core competencies, without the hassle of dealing with large hardware platforms, many software vendors, network providers and extranets, maintaining relationships with a range of exchanges, sourcing internal market data expertise and performing 24 hours a day, seven days a week system monitoring.

SuperFeed was developed to meet client demand for a single-vendor solution that provides cost-efficient access to many markets via ultra-low-latency connections over the high-reliability, low-latency IP access network that supports the securities industry, the Secure Financial Transaction Infrastructure® (SFTI®) network, or co-location. SuperFeed provides both normalized and value-added market data over two feature-rich, unified API interfaces: Middleware Agnostic Messaging API™ (MAMA™) and Middleware Agnostic Market Data API™ (MAMDA™). This provides both backwards compatibility with in-house ticker plants and a future-proof interface designed to support changes and new content.

With a track record of high performance, flexibility and reliability, we provide a platinum-level service around connectivity, hardware, operating system management, software management, support desk, Entitlements Vendor of Record and reporting. Our service is guaranteed to meet the market data needs of single user platforms, high frequency trading applications and matching engines.

Alongside cutting edge technology, SuperFeed is monitored 24 hours a day, seven days a week by a highly specialized global managed service unit. This service ensures maximum reliability and service standards across all site locations in North America, Europe and Asia.

### 1.1 TECHNOLOGY OVERVIEW

SuperFeed runs on the same mature, industry-leading software that NYSE Technologies' customers use today – feed handlers (software that handles incoming market data and publishes it in a normalized form), MamaCaches (an application that can be used for distribution, caching and bridging of market data to additional sites or client groups, while protecting the core infrastructure), SuperBook™ for equities order book aggregation (an aggregated order book that covers multiple liquidity centers) and Data Administration and Reporting Tools™ (DART™) Entitlements. The key technology differentiator is NYSE Technologies Data Fabric™, our premier middleware. With implementations both on InfiniBand and 10 gigabit Ethernet platforms, Data Fabric ensures double-digit microsecond performance through data feed normalization, providing our clients with a solid competitive advantage.

SuperFeed is a hosted ticker plant to which clients can connect to access market data from a number of different data sources. API access allows clients to make selective subscriptions, minimizing bandwidth use and costs incurred. Subscriptions are interactive and can be created and removed intraday. SuperFeed controls the entitlements for each market data feed source so that access controls are in place for all disseminated data.

To connect to SuperFeed, clients must use MAMA and/or MAMDA to interact with the platform. Each end user is explicitly controlled and reported through DART Entitlements. Clients can connect as data feed recipients or as controlled device systems. As controlled device users, clients are prohibited from market data redistribution by signing a series of exhibits and provisions that exclude the ability to redistribute market data, including signing flow down legal provisions for each exchange that contributes to SuperFeed.

Data feed recipients receiving market data via SuperFeed, must file the appropriate subscriber agreements for each location receiving data feed access to SuperFeed. Clients are responsible for reporting any market data redistribution and usage within their organization or externally.

Entitlements reports of systems connecting to SuperFeed are sent monthly to TCB Data in VRXML format. These reports include a breakdown of all clients plus a list of data feeds and controlled devices that have been entitled to receive market data from SuperFeed.

Clients can select from a broad range of market data feeds from various exchanges using SuperFeed, including NYSE, NYSE Amex, NYSE Arca, NASDAQ, BATS, Direct Edge, CME, CBOE, and ISE. From within these exchanges, they can purchase individual market data feed access covering equities data and derivatives, such as,

options and futures. These data feeds often include quotes, trades, depth of book, imbalance and security status information.

### **1.1.1 Platform Flexibility**

Today's market data challenges are not solely focused on market fragmentation, but also technology fragmentation. Organizations often find themselves with two or more market data systems powering their organization for various business functions. SuperFeed was designed to meet the needs of a hybrid organization, catering for direct connection GUIs, high frequency algorithmic systems and/or data feed customers.

SuperFeed supports both direct application connections (Figure 1) and connections using the Super Feed feed handler (Figure 2).

Figure 1 Platform Architecture Diagram – Direct Connections

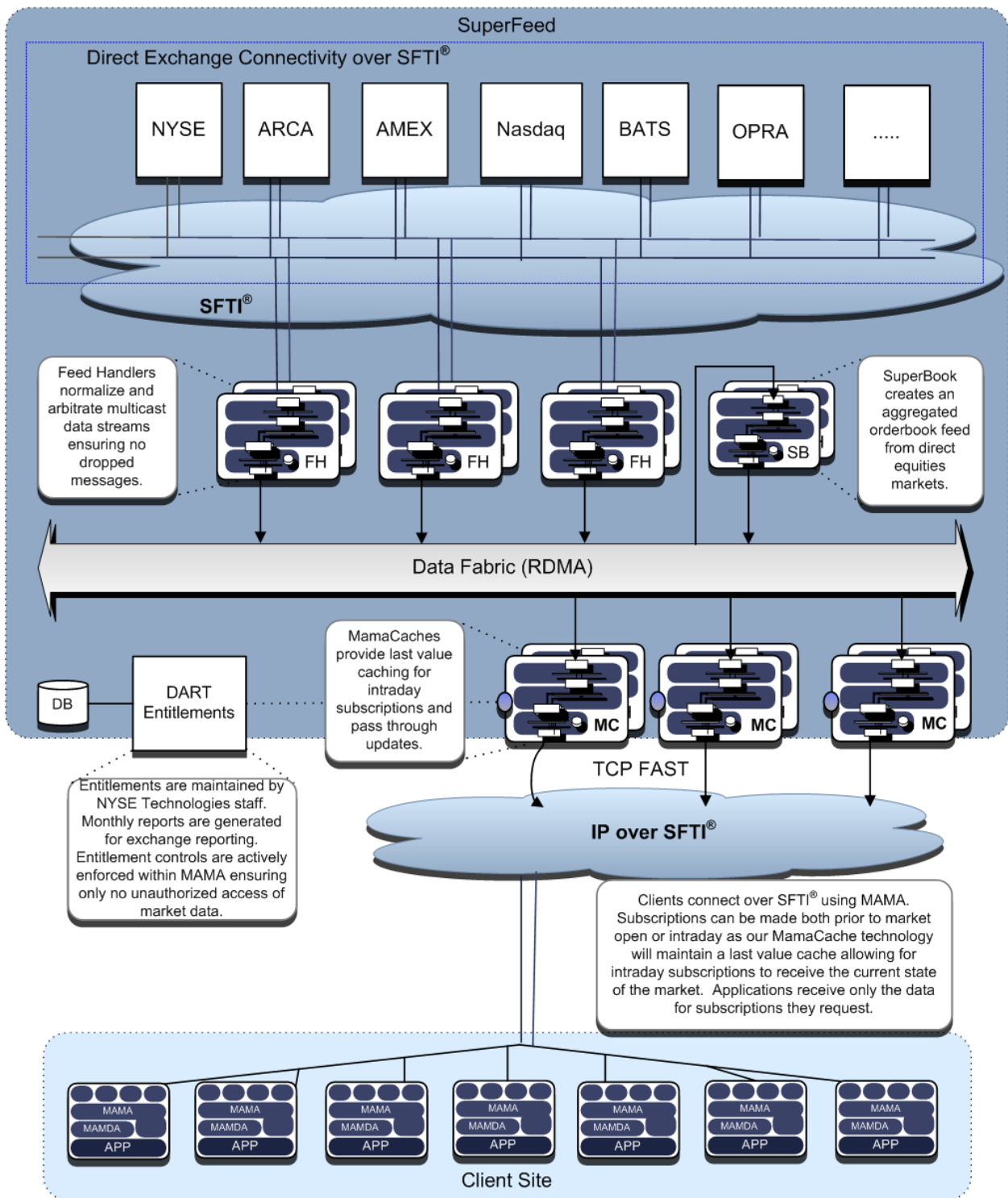
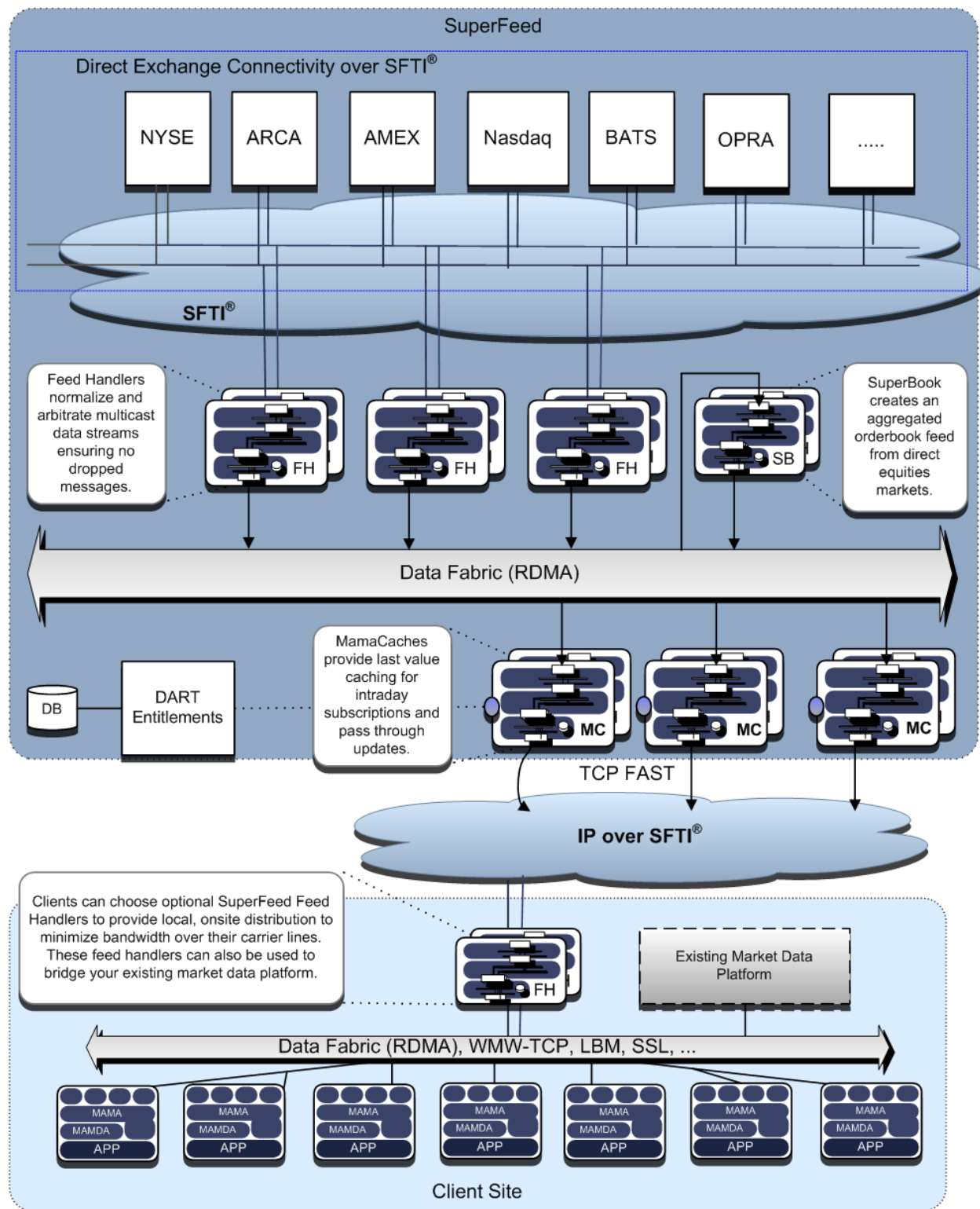


Figure 2 Platform Architecture Diagram – SuperFeed Feed Handler Local Distribution



1.1.2 Data Sourcing and Entitlements

SuperFeed acquires raw market data through the SFTI network from various markets, including NYSE Euronext markets and non-NYSE Euronext markets. Raw data is normalized through the hosted ticker plant and client access is allowed through the MamaCache layer. Clients connect using MAMA or MAMDA only, via unicast connections and subscribing to specific content sets.

Entitlements controls are served by the DART Entitlements which employs both active and passive entitlements, ensuring that users only access data to which they are entitled.

## 1.2 APIS: MAMA AND MAMDA

Clients connecting to SuperFeed must use NYSE Technologies' APIs, MAMA and MAMDA, to access market data. Since the market data feed is delivered as an API feed, no other alternative is available for making subscriptions and receiving market data to an application.

Clients can either use our sample applications, built using MAMA and MAMDA, or they can write/integrate their own application and presentation logic on top of MAMA and MAMDA. Embedded into MAMA are all the communication, subscriptions, entitlement control, network-level protocol libraries for connection and message-handling logic. The client has no way to alter the underlying classes that handle these mechanisms: the client can only provide the configuration necessary for the application to work and business logic needed to process each incoming message received by the API.

MAMA is a high-performance client-side integration layer that interfaces with a variety of messaging middleware systems. It provides an ultra-lightweight, future-proof integration layer that provides users with the flexibility to embrace new middleware technologies and applications as the market innovates. With its ability to process millions of messages per second with nanosecond latencies (middleware dependent), and capable of sustaining hundreds of thousands of simultaneous subscriptions, MAMA provides a compelling alternative to using native middleware APIs directly.

Developers can create robust messaging-based applications without coupling to a specific middleware vendor. Code changes are not required when switching underlying middleware vendors, allowing greater flexibility of application deployment while significantly reducing the business risk of switching middleware platforms.

Traditional APIs used for market data distribution, including MAMA, provide anonymous field and/or record based interfaces. These APIs require applications to iterate over fields and provide internal logic to determine what type of message the collection of fields represents. Hence they are adequate for certain types of applications that do not need to understand the context of the data being consumed. Other applications, such as program trading, tick capture, analytical and smart order routing applications need a deeper understanding of the meaning and context of each field in the message as well as the type of message the update represents. MAMDA is a higher-level API that provides a much more meaningful and context-sensitive programming interface than simple field-based APIs.

MAMDA is a wrapper around MAMA that extends a layer of market data aware objects built on top of MAMA. Effectively, MAMDA extends MAMA, creating an object-oriented API that intrinsically understands market data constructs, such as, quotes, trades, imbalance data, cancels / corrections, order books, order book deltas, news and various other objects within the realm of market data. MAMDA also performs caching so that application developers can easily retrieve last known values for data fields.

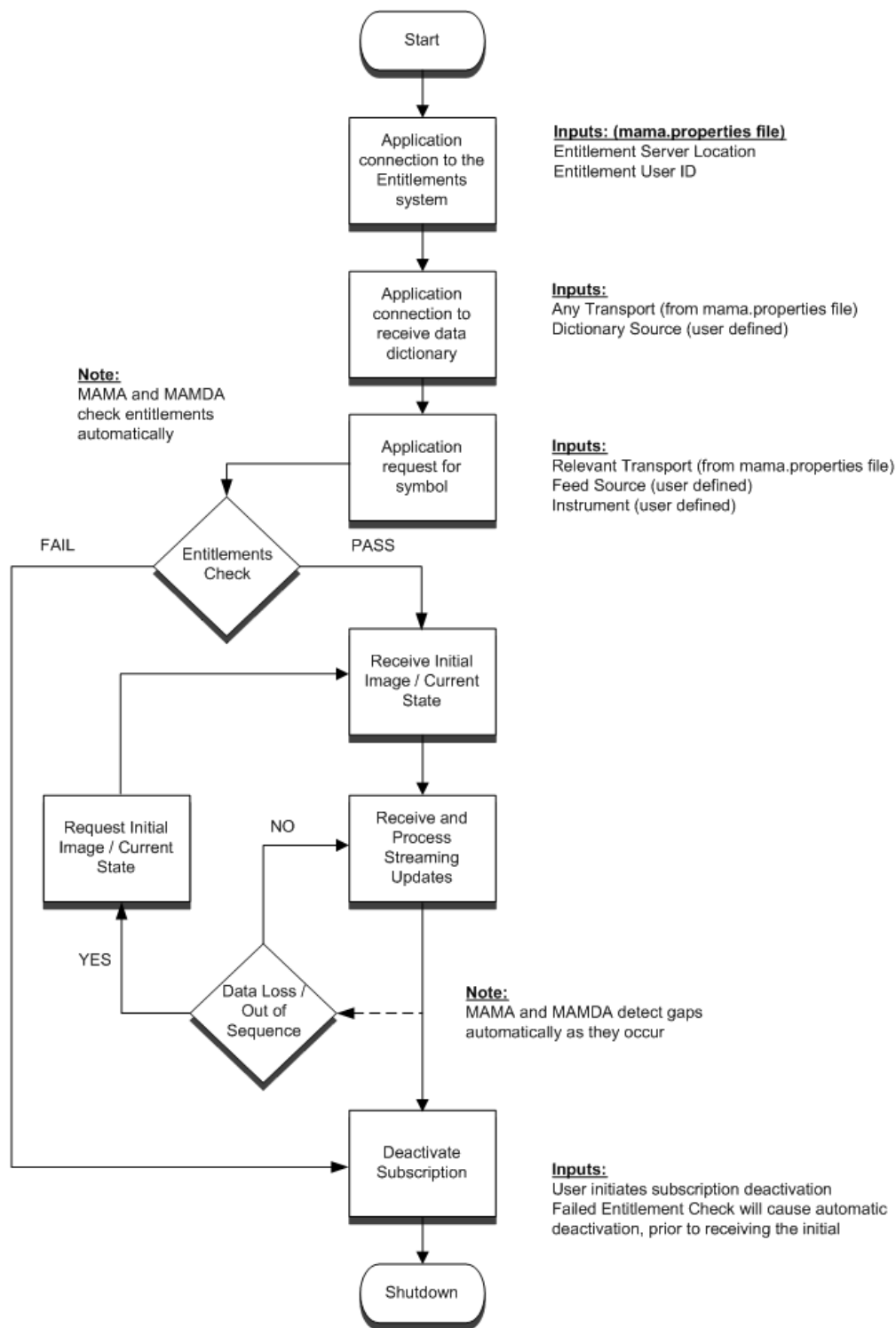
Classifying market data messages extends beyond determining the messages' high level types: quotes, trades, order book updates, security status updates, order imbalance data, etc. Applications must further categorize these messages into subtypes: regional verses consolidated quotes, regional verses consolidated trades, trade cancels/errors/corrections, out of order trades and so on. The field based approach to accessing market data demands a great deal of complex logic to correctly classify incoming data. MAMDA augments MAMA with a rich set of market data related data structures that address the shortcomings of field-based APIs. The object-oriented MAMDA (available in C++, Java and .Net), performs the tedious work of deciphering field based data from MAMA into various types of messages and provides convenient interfaces for applications to process this data.

MAMDA also provides object models for rich, structured market data types such as order books and options chains, removing the need for application developers to incur the overhead of creating and maintaining these complex structures.

For guidance on developing applications with MAMA and MAMDA, see the MAMA and MAMA developer documentation (refer to Further Information section).

The lifecycle of a MAMA or MAMDA application is shown in Figure 3 MAMA and MAMDA Application Lifecycle.

Figure 3 MAMA and MAMDA Application Lifecycle



1.2.1 Entitlement Checking

When a client instantiates a subscription:

- The API checks to see if the source of the subscription is allowed – e.g. NASDAQ would be a source. If this succeeds, the subscription request is sent to the SuperFeed publisher. If this fails, the subscription is denied and the application is informed that the application is not entitled for the requested data.

- If the SuperFeed Publisher receives a subscription request it will send back the current state of the security requested, and streaming updates over TCP.
- Upon receipt of any market data message, the API checks the entitlement code embedded with each update. If the entitlement code exists within the list of allowed products, the API will pass on the data as an event to the application logic. If the entitlement code doesn't exist within the list of allowed products, the subscription for that instrument is terminated by the API and no further updates are sent by SuperFeed.

### 1.3 GLOBAL MARKET DATA ADMINISTRATION TEAM SERVICES

As part of the SuperFeed service, the Global Market Data Administration team performs the following functions listed in Table 1.

**Table 1 Global Market Data Administration Team Services**

On Behalf of SuperFeed Clients	On Behalf of NYSE Technologies
<ul style="list-style-type: none"> <li>• <b>Ascertain exchange requirements</b></li> <li>• <b>Obtain exchange agreements</b></li> <li>• <b>Assist with the completion of any exchange exhibits, questionnaires</b></li> <li>• <b>Submit request for feed authorization</b></li> <li>• <b>Assist with any reporting requirements</b></li> </ul>	<ul style="list-style-type: none"> <li>• Obtain exchange authorization for new clients and / or new products</li> <li>• Coordinate exchange approval with DART entitlement</li> <li>• Submit monthly exchange reporting</li> <li>• Facilitate exchange payments</li> </ul>

### 1.4 CONNECTIVITY

Connectivity to hosted SuperFeed requires connectivity to the SFTI network. For more information on SFTI Global Connectivity Services please see <http://www.nyse.com/technologies>.

All data products on the SuperFeed platform are currently disseminated on TCP/IP connections, allowing for on-demand subscriptions throughout the day. Intraday subscriptions are handled by providing the current state snapshot of the market for the requested instrument, followed by the streaming updates.

### 1.5 MARKET COVERAGE

For a list of available feeds, see the [SuperFeed Market Coverage](#) document.

## 2 PLATFORM COMPONENTS

The following sections describe the components shown in Figure 1 Platform Architecture Diagram – Direct Connections and Figure 2 Platform Architecture Diagram – SuperFeed Feed Handler Local Distribution.

### 2.1 FEED HANDLERS

The SuperFeed market data platform contains a tier of feed handler technology, which handles the processing and normalization of raw market data feeds covering various markets and asset classes. The primary functions of the feed handlers are described below:

- a. Process raw market data packets according to each data feed specification
- b. Maintain and build last value cache data for record and order book data received from the data feed
- c. Normalize the incoming market data to a consistent message format compatible with the MAMA and MAMDA APIs
- d. Assign update types to the normalized data for efficient callback access
- e. Republish the normalized data onto a market data message bus, known as the Data Fabric

The feed handlers, which run as a hosted service within the SuperFeed environment, are managed by NYSE Technologies Managed Services and Support staff. Clients connecting to SuperFeed never interact directly with the feed handlers; they interact solely through MamaCache. In addition, all required changes due to operations, exchange specifications and capacity are handled by NYSE Technologies staff. Clients will be notified only in the event that a change affects their API functionality or requires new configuration changes on their side.

Each feed handler is individually configured to appropriately process and normalize market data for a specific data feed. Links to the feed handler mapping guides are detailed in the SuperFeed Market Coverage document.

For redundancy, each feed handler has both a primary and secondary instance. Ethernet and InfiniBand networking components also have fully redundant switching.

### 2.2 SUPERBOOK

SuperBook is an order book aggregation engine that combines various depth-of-book market data feeds into one depth-of-book feed. It currently aggregates only US equities feeds and EU cash markets. Individual feeds are given a participant ID so that the end user can distinguish which feed is providing a specific order book entry. SuperBook provides a combined view of liquidity across multiple markets on a per instrument basis, offering a consolidated view of market depth.

SuperBook is owned and operated as a fully managed service by NYSE Technologies staff. Functionally, this technology is operated like a feed handler for a synthetic feed derived from multiple direct exchange feeds. In addition, all required changes due to operations, exchange specifications and capacity are handled by NYSE Technologies staff. Clients will be notified only in the event where a change affects their API functionality or requires new configuration changes on their side.

As with the feed handlers, SuperFeed has both primary and secondary instances for redundancy. Ethernet and InfiniBand also have fully redundant switching.

#### 2.2.1 Entry IDs

SuperFeed uses a one or two letter code to identify the Entry ID source feed for the order book entry in SuperBook, the aggregated order book product.

SuperBook provides the aggregated order book from each of the available depth feeds and fills out the BBO (best bid and offer) entry in the consolidated feeds (CQS or UQDF) for each participant exchange in which no direct feed is available for full depth. The CQS feeds cover Tape A and B securities, while Tape C is covered in the UQDF feed.

No trade or security status data is available in SuperBook.

#### 2.2.2 Entry ID Example

Entry IDs are used in SuperBook to determine the source feed for the entry in the aggregated book. Clients must map participant IDs to the appropriate feed source in order to properly determine which market and feed the order book entry comes from. An example of how the Entry IDs are provided is shown in Figure 4 Entry IDs in SuperBook.

Mappings for the IDs are in Table 2 Direct Feed Entry ID codes to Table 5 Consolidated Entry IDs for UQDF/UTDF (NASDAQ).

**Figure 4 Entry IDs in SuperBook**

		IBM					
ID	Time	Size	Price	Price	Size	Time	ID
Q	17:38:21.003	300	100.8000	100.8300	1500	17:38:14.613	P
N	17:38:15.139	100	100.8000	100.8300	500	17:38:21.695	Q
C1	17:38:14.613	200	100.8000	100.8300	700	17:38:21.575	N
FLOW1	17:38:20.686	100	100.8000	100.8300	300	17:38:14.611	Z
Q	17:38:21.003	400	100.7900	100.8300	100	17:38:18.036	I1
P	17:38:21.004	200	100.7900	100.8300	200	17:38:21.014	B1
Z	17:38:21.003	200	100.7900	100.8400	700	17:38:18.414	Q
B1	17:38:14.731	100	100.7900	100.8400	600	17:38:19.252	P
N	17:38:21.883	100	100.7900	100.8400	300	17:38:14.608	Z
Q	17:38:16.188	800	100.7800	100.8400	200	17:38:14.719	C1
N	17:38:14.623	100	100.7800	100.8400	100	17:38:18.416	N
P	17:38:14.802	600	100.7800	100.8500	400	17:38:21.743	Q
Z	17:38:14.608	400	100.7800	100.8500	300	17:38:14.325	Z
Q	17:38:14.708	1000	100.7700	100.8500	600	17:38:14.022	N
P	17:38:14.609	100	100.7700	100.8600	2300	17:38:21.745	Q
Z	17:38:14.608	200	100.7700	100.8600	100	17:38:13.997	P
N	17:38:16.033	500	100.7700	100.8600	300	17:38:14.142	Z
Q	17:38:21.242	500	100.7600	100.8600	1700	17:38:14.613	N
P	17:38:14.328	900	100.7600	100.8700	1800	17:38:16.563	Q
Z	17:38:14.149	300	100.7600	100.8700	2700	17:38:15.067	N
N	17:38:21.010	200	100.7600	100.8700	200	17:38:08.451	Z
I1	17:37:57.563	100	100.7600	100.8700	100	17:38:14.382	P
P	17:38:14.615	1400	100.7500	100.8800	600	17:38:18.175	Q
Q	17:38:21.243	2200	100.7500	100.8800	600	17:38:15.158	P
N	17:38:21.016	2500	100.7500	100.8800	1100	17:38:20.533	N
Q	17:38:18.194	1500	100.7400	100.8900	400	17:38:08.414	Q
N	17:38:15.534	3800	100.7400	100.8900	3000	17:38:14.059	N
Z	17:38:15.145	1300	100.7400	100.8900	2300	17:38:21.592	P
N	17:38:20.580	3700	100.7300	100.8900	800	17:38:16.836	Z
Q	17:38:14.107	400	100.7300	100.9000	1400	17:38:14.729	Q
P	17:38:15.142	300	100.7300	100.9000	3200	17:38:18.528	N
P	17:38:10.813	100	100.7200	100.9000	400	17:38:22.012	P
Q	17:38:18.525	1200	100.7200	100.9000	200	17:38:16.838	Z
N	17:38:09.570	500	100.7200	100.9100	700	17:38:14.616	Q
N	17:38:20.534	300	100.7100	100.9100	1300	17:38:14.619	N
Q	17:38:14.607	700	100.7100	100.9100	1300	17:38:22.008	P

**Table 2 Direct Feed Entry ID codes**

Value	Description
P	ArcaBook
N	NYSE OpenBook Ultra
Z	BATS Depth of Book
A	AMEX OpenBook Ultra
Q	TotalView 3.1
IA	Direct Edge A / ISE
IX	Direct Edge X / ISE

**Table 3 Consolidated Entry IDs from CQS/CTS for NYSE Listed Securities**

Value	Description
A1	NYSE AMEX on CQS
B1	NASDAQ OMX BX on CQS
C1	National Stock Exchange on CQS
D1	FINRA on CQS
I1	International Stock Exchange on CQS
J1	Direct Edge A on CQS
K1	Direct Edge X on CQS
M1	Chicago Stock Exchange on CQS
N1	New York Stock Exchange on CQS
FLOW1	Lava Flow ECN (ADF) on CQS
P1	NYSE Arca on CQS
T1	NASDAQ OMX on CQS
W1	CBOE Stock Exchange on CQS
X1	NASDAQ OMX PHLX on CQS
Y1	BATS Y Exchange on CQS
Z1	BATS Exchange on CQS

**Table 4 Consolidated Entry IDs from CQS/CTS for AMEX and ARCA Listed Securities**

Value	Description
A2	NYSE AMEX on CQS
B2	NASDAQ OMX BX on CQS
C2	National Stock Exchange on CQS
D2	FINRA on CQS
I2	International Stock Exchange on CQS
J2	Direct Edge A on CQS
K2	Direct Edge X on CQS
M2	Chicago Stock Exchange on CQS
N2	New York Stock Exchange on CQS
FLOW2	Lava Flow ECN (ADF) on CQS
P2	NYSE Arca on CQS
T2	NASDAQ OMX on CQS
W2	CBOE Stock Exchange on CQS
X2	NASDAQ OMX PHLX on CQS
Y2	BATS Y Exchange on CQS
Z2	BATS Exchange on CQS

**Table 5 Consolidated Entry IDs for UQDF/UTDF (NASDAQ)**

Value	Description
A3	NYSE AMEX on UQDF
B3	NASDAQ OMX BX on UQDF
C3	National Stock Exchange on UQDF
D3	FINRA on UQDF
I3	International Stock Exchange on UQDF
J3	Direct Edge A on UQDF
K3	Direct Edge X on UQDF
M3	Chicago Stock Exchange on UQDF
N3	New York Stock Exchange on UQDF
P3	NYSE Arca on UQDF
Q3	NASDAQ OMX on UQDF
W3	CBOE Stock Exchange on UQDF
X3	NASDAQ OMX PHLX on UQDF
Y3	BATS Y Exchange on UQDF
Z3	BATS Exchange on UQDF

### 2.3 MAMACACHE

Clients connecting to SuperFeed interact only with MamaCache which handles caching and distribution. MamaCaches are set up in a primary and secondary configuration for redundancy, and handle one or many upstream data feed sources depending on how they are load balanced. Each connecting client connects via TCP / IP, using the MAMA API to interactively make subscriptions to MamaCache for market data. The connection lifecycle is shown in Figure 3 MAMA and MAMDA Application Lifecycle.

The MamaCache technology, through its ability to send current state of an instrument along with streaming updates, can satisfy intraday subscriptions. This functionality allows users to subscribe intraday and to recover from missed messages by re-requesting current state (known as a “recap”). MAMA and MAMDA automatically handle intraday subscriptions and the detection of data loss, and will request recaps to MamaCache without the need for user interaction.

For compressed feeds from SuperFeed, FAST compression is applied in the MamaCache tier, providing a reduced-bandwidth feed that often has several messages batched within a packet. Clients can simultaneously process FAST compacted messages and non-compacted messages as the handling of these messages is performed entirely in the MAMA layer, requiring no development changes for users of MAMA or MAMDA.

Clients can also destroy subscriptions interactively, ensuring that no more data is sent to the client for a given instrument from the MamaCache. This is useful for applications that only require periodic market data updates for particular instruments. Please refer to the MAMA and MAMDA developer’s guides for more information on this feature.

In the event of a primary MamaCache failure, the secondary server will take over publishing, broadcasting recaps to all connected clients for the symbols they are subscribed to. The secondary server will subsequently send streaming updates so that the end user clients do not have any disruption of services. This is handled transparently in MAMA. In MAMDA, callback structures are present to allow application logic to be processed in the event of these fault tolerant take-overs. Please refer to the MAMA and MAMDA developer’s guides for more information on this feature.

MamaCache is operated as a fully managed service by NYSE Technologies staff. All required changes due to operations, exchange specifications, and capacity are handled by NYSE Technologies staff. Clients will be notified only in the event where a change affects their API functionality or requires new configuration changes on their side.

## 2.4 SUPERFEED FEED HANDLER

The SuperFeed feed handler is an optional component that may or may not be required by a client using SuperFeed. The SuperFeed feed handler acts as a locally-deployed proxy, allowing for all clients to request data from it and making all upstream requests to SuperFeed.

Using technology similar to MamaCache, this feed handler caches the last value for each requested symbols and services downstream clients directly, while interacting with MamaCaches hosted in the SuperFeed environment.

This technology allows for three main functions:

- a. Allows for TCP bandwidth aggregation upstream and data fan-out to local clients, such that no duplicate data is sent from SuperFeed to a remote site
- b. Allows for bridging SuperFeed data streams onto a market data platform running Data Fabric, NYSE Technologies TCP Middleware, LBM, SSL and all other middlewares supported by NYSE Technologies
- c. In the future, when multicast feeds are available on the SuperFeed platform, the SuperFeed feed handler will be required to consume the multicast streams and bridge onto a middleware existing on the local client site

Client interaction between a MamaCache and SuperFeed feed handler is interchangeable, allowing for a 100% flexible design and ease of implementation of the feed handler.

SuperFeed feed handlers are operated and configured by the client. Managed Services are available for these software components. However these are not included as part of SuperFeed standard commercial licensing. It is also the responsibility of the client to maintain all changes for this software, including configuration and capacity management of hardware. For bandwidth estimates for SuperFeed, see <http://www.nyxdata.com/capacity/>.

## 2.5 ENTITLEMENT SERVER

Entitlement management and reporting is part of the SuperFeed service. All connecting clients must use a specific user ID that has been created and provisioned for fee-liable data. The entitlement server is used by SuperFeed Administrative staff to maintain a database of entitlements used for both active entitlements controls and for monthly reporting.

All connecting MAMA or MAMDA clients must successfully authenticate to the Entitlement Server proxy, known as a "Site Server" to identify themselves and to receive a download of authorized entitlements. No market data subscriptions will succeed unless proper entitlements are obtained by the MAMA API from the Entitlements server.

NYSE Technologies staff maintains this system and the monthly reporting. Changes in entitlements can be requested using the Entitlements change request form. Note that changes will only be approved for services commercially purchased from SuperFeed and for exchange feed access that is already approved by the disseminating exchange. All requests for new feed services need to be requested from your NYSE Technologies Sales Representative, [NYSE-Technologies-Sales@nyx.com](mailto:NYSE-Technologies-Sales@nyx.com).

## 2.6 MAMA AND MAMDA APIS

MAMA and MAMDA are the development framework on which to build market data applications. These APIs provide a rich level of functionality allowing a developer to rapidly integrate applications to using SuperFeed for market data.

The APIs are used to interactively subscribe to the SuperFeed platform for both streaming updates and snapshot data.

MAMA and MAMDA are described in Section 1.2, APIs: MAMA and MAMDA.

### 2.6.1 Development Tips and Guidelines

You should consider the following areas when building MAMA and MAMDA applications.

1. Assign an application name to running instances. This can help determine what logical process is connecting and is generally very useful if a specific functional process has to be identified. `setApplicationName()`. Please refer to the MAMA and MAMDA developer's guides for more information on this feature.
2. Avoid making duplicate subscriptions within one running instance to avoid forcing applications to process more data than is necessary.
3. The javadocs list and explain the expected return values for accessor methods. Please refer to this documentation for proper handling of normalized field values.

4. If only a snapshot of data is needed, it is best to create a subscription type of Snapshot. Please refer to the MAMA and MAMDA developer's guides for more information on this feature.
5. Applications should have administrative features, such as one that can dynamically alter the log level of the application to capture valuable information for troubleshooting. Applications should also have administrative methods that allow clients to intervene or shut down the application if it is performing poorly.
6. Applications should use event queue monitoring, setting high and low water marks. It is good practice to at least log this information in the case it needs to be analyzed at a later date. See the Mama Developer's guide for how to set watermarks for event callbacks or create an ad-hoc query to the size of the event queue.
7. Applications should use appropriate throttling per transport. The API allows requests to be throttled in a defensive way to prevent the application from requesting too much data at once. There are server side controls for this, however its good practice to define these in applications as server controls are general controls that apply to every application connecting. These are covered in the MAMA developers guide, but some examples are:

```
transport.setOutboundThrottle(MamaThrottleInstance.INITIAL_THROTTLE, 100);
//Sets the max per second rate of throttling subscription requests

transport.setOutboundThrottle(MamaThrottleInstance.RECAP_THROTTLE, 50);
//Sets the max recap request rate

transport.setOutboundThrottle(MamaThrottleInstance.DEFAULT_THROTTLE, 150);
//Sets total aggregate max throttle of the two above
```

In this example, if there was one transport for NYSE equities data, it would throttle at 100 new subscription requests per second. E.g. if it needed to make 3000 subscriptions, it would take 30 seconds to do so.

Recaps occur when the application receives out of sequence data, or if data is marked as stale. A recap is essentially a request for a new initial image, as the API believes it may be out of sync with the publisher. Its best to define a maximum on how many recaps the API will request. In general, this should never exceed the max initial request rate. In the worse case scenario, if you were subscribing to 3000 symbols, and all of them required recaps, a recap rate of 50 per second would mean a 60 second recovery time.

8. It is useful to embed simple data quality checks in MAMA and MAMDA applications. Some conditions that may be monitored to send alerts to the log file are:
  - No data received on an entire transport for X seconds
  - No updates received in X seconds for a symbol
  - Locked or crossed indication e.g. Bid > Ask. Note that not all locked or crossed market conditions are indicative of an issue. Some exchange feed providers specify this condition to be within normal operating conditions.
  - No open prices received for a symbol, or no closing price received
9. The Web Issue Management (WIM) system should be used to report issues with or ask questions about MAMA and MAMDA. WIM is described in 3.1, Web Issue Management System.
10. SuperFeed only sends field updates for values that have changed. To retrieve values from fields that don't change on every tick, please use the Quote and Trade Listener objects, as they maintain all the cached values of data fields. For more information, please refer to the MAMDA Developer's guide and/or API documentation.

### 3 SUPPORT

Issues with SuperFeed are reported and tracked using Web Issue Management (WIM), telephone contact and email contact. For email and telephone contact, please contact the Global Service Desk using the details at the start of this document.

#### 3.1 WEB ISSUE MANAGEMENT SYSTEM

NYSE Technologies uses WIM to track all questions, issues and tasks. WIM is available at <http://issues.wombatfs.com/>. Each designated user will be assigned a WIM account. If further login details are required please contact NYSE Technologies using the details in Contact Information..

When an issue is created the reporter automatically receives email notifications of any updates made by NYSE Technologies. A user can also receive updates for issues reported by a third party by following these steps.

1. In WIM, go to the issue to be followed.
2. Click "Watch it" in the left hand column (towards the bottom of the "Operations" list).

WIM uses a 5 level scale to classify issues. These are described in Table 6 WIM Classifications.

**Table 6 WIM Classifications**

Classification	Description	Examples
<b>Blocker</b>	One or more production components of the system are unavailable or the service has been degraded to a point where the component(s) are unusable	A process has crashed The TCP connection to the exchange has dropped and fault tolerance mechanisms have not rectified the issue.
<b>Critical</b>	One or more production components of the system are defective but still usable,	The primary multicast line stops receiving data but the backup is functioning. A process has crashed but fault tolerance has rectified the issue.
<b>Major</b>	A component of the system is defective but not affecting business, or an immediate workaround is available.	An incorrect field value that may be rectified using a wadmin command.
<b>Minor</b>	A component of the system is not operating as desired but having limited effect on production usage.	An incorrect field value on a symbol which is not being traded.
<b>Trivial</b>	A component of the system is not operating as desired but having no effect on production usage	A spurious log message in the log file.

#### 3.2 TELEPHONE

Clients can raise issues by telephoning the number in Contact Information. A call answering service is used to capture clients' contact details and route the call to the relevant NYSE Technologies personnel. This is supplemented by mobile escalation channels.

Telephone issues will be raised within WIM at earliest convenience for tracking purposes, and assigned a classification as described.

Clients should use the telephone helpdesk for any issue causing production service degradation (classified as "Blocker" or "Critical" as described in Table 6 WIM Classifications). Except in the case of a technology issue, a service consultant will remain on the line until a satisfactory resolution is in place, or the client is content that the issue can be completed with a callback when resolved.

#### 3.3 EMAIL

Clients can raise issues by emailing [service.desk@nyx](mailto:service.desk@nyx). This procedure should only be used when access to WIM system is unavailable. NYSE Technologies will raise the issue in WIM for tracking purposes. A classification of 'Major'

will be assumed if no classification is clear from the email. This classification may be modified subsequently at the client's request in accordance with the prescribed definitions.

### **3.4 MONITORING**

When the monitoring systems alarms to a potential issue NYSE Technologies managed service team will investigate immediately. An email notification will be sent to clients to a pre-agreed mailing list, alerting clients to the alarm and that investigation is underway. A WIM issue will be logged at the earliest opportunity and affected clients will be sent the WIM ID for tracking.

### **3.5 ESCALATION**

To assist with resolution, the support team will escalate to NYSE Technologies executives if:

- an issue is not resolved; or
- a workaround is not in place; or
- a solution is not in progress

within the timescales listed in the client's service-level agreement (SLA). Clients may request escalation prior this if deemed appropriate.

## 4 SYSTEM REQUIREMENTS AND RECOMMENDATIONS

For detailed system requirements, consult the MAMA and MAMDA developers guides and the Data Fabric documentation.

Each node (whether connecting to SuperFeed over the SFTI network by co-location) should meet the following requirements.

- The operating system should be one of the following.
  - Red Hat Enterprise Linux 4 Update 5 (RHEL 4U5) or higher
  - SUSE Enterprise Linux 10 Service Pack 1 (SLES 10SP1) or higher
- All of these packages are required.
  - gcc
  - pci-utils
  - kernel-sym

All connecting clients are required to conform to our standard `sysctl.conf` settings for Linux:

```
# NYSE Technologies suggested buffer settings
net.core.rmem_max = 16777216
net.core.wmem_max = 16777216
net.core.rmem_default = 8388608
net.core.wmem_default = 8388608
net.core.netdev_max_backlog = 3145728
net.ipv4.tcp_window_scaling = 0
net.ipv4.route.secret_interval = 86400
net.ipv4.tcp_rmem = 262144 4194304 16777216
net.ipv4.tcp_wmem = 262144 4194304 16777216
net.ipv4.tcp_mem = 262144 4194304 16777216
net.ipv4.tcp_no_metrics_save = 1
net.ipv4.tcp_sack = 0
net.ipv4.tcp_dsack=0
net.ipv4.tcp_timestamps=0
```

In addition we recommend disabling all of the following processes. Processes with an asterisk (\*) are considered to be highly detrimental to production operating environments, causing persistent and frequent latency issues :

- iptables \*
- cpuspeed \*
- auditd
- bluetooth
- cups-config-daemon
- gpm
- httpd
- irda
- irqbalance \*
- kudzu
- lisa
- multipathd
- NetworkManager
- nscd
- postfix

- snmpd
- winbind

#### 4.1 WINDOWS TUNING PARAMETERS

1. Tcp1323Opts

HKLM\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters

This registry entry may need to be created if it does not exist in Parameters. The registry entry would need to be REG\_DWORD with value "0"

2. TcpWindowSize

HKLM\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters

The binary value of this parameter should be changed to "16777216"

3. GlobalMaxTcpWindowSize

HKLM\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters

This registry entry may need to be created if it does not exist in Parameters. The registry entry would need to be REG\_DWORD with decimal value "16777216"

4. DefaultReceiveWindow

HKLM\SYSTEM\CurrentControlSet\Services\Afd\Parameters

This registry entry may need to be created if it does not exist in Parameters. The registry entry would need to be REG\_DWORD with decimal value "16777216"

5. DefaultSendWindow

HKLM\SYSTEM\CurrentControlSet\Services\Afd\Parameters

This registry entry may need to be created if it does not exist in Parameters. The registry entry would need to be REG\_DWORD with decimal value "16777216"

Note: If any of these registry keys do not exist, they must be created in the specified folder specified. "HKLM" denotes HKEY\_LOCAL\_MACHINE

#### 4.2 MAMA AND MAMDA REQUIREMENTS

MAMA and MAMDA are available for C++, Java (JNI) and .Net. MAMA is also available for C.

MAMA and MAMDA are available on the following platforms:

- Solaris 8 and up (SPARC and x86)
- Windows 2000/XP (VC 7/8) (Visual Studio 2003/2005)
- Linux 2.4 kernel (libc 2.2.4 and above, gcc 2.9.6 and above) (x86/x86\_64)
- Linux 2.6 kernel (libc 2.3.3 and above, gcc 3.3.3 and above) (x86/x86\_64)

#### 4.3 CAPACITY

Access to the SuperFeed hosted ticker plant requires connectivity to the SFTI network, either through a customer's SFTI circuits (which generally are 100 mbit, 1 Gbit, or 10 Gbit), or through an extranet which has connectivity to the SFTI network. All connections are TCP, and as such, it is the customer's responsibility to manage bandwidth usage and allocation effectively. Where available, SuperFeed will compress market data streams using FAST.

SuperFeed delivers data through an interactive API, allowing for only market data that is requested to be received. Duplicate subscriptions can be made, however this will result in additional bandwidth being used by the consumer(s).

Guideline figures are available at <http://www.nyxdata.com/capacity>.