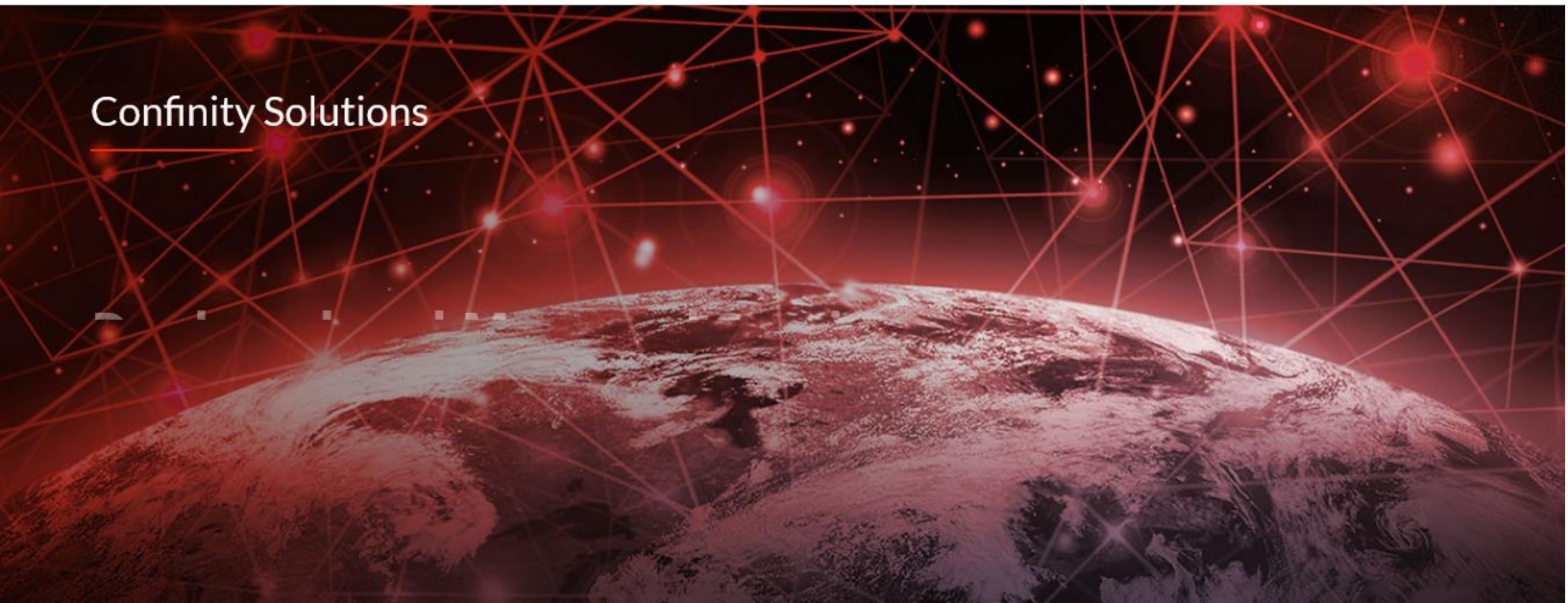




# Confinity Solutions Market Data Distribution System

*Components, capabilities and advantages*









## Contents

- 2 Executive summary
- 2 Overview of Confinity Market Data Distribution System
- 6 General architecture
- 8 In depth: Product components
- 15 Flexibility and scalability
- 16 Ultra-low latency, high throughput messaging transport
- 16 Compatibility with existing systems
- 17 Designed for mission-critical operations
- 19 Advantages of CMDS
- 20 For more information

## Executive summary

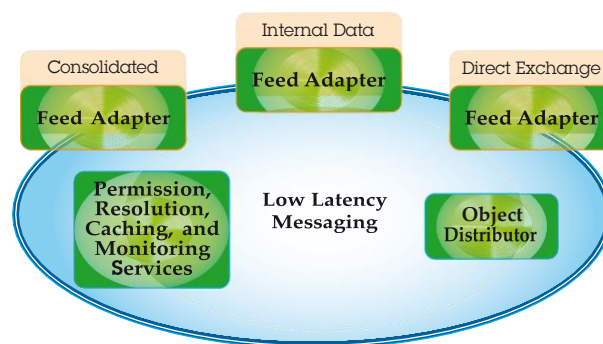
Confinity Market Data Distribution System (CMDS) for Financial Markets is a robust market data platform providing ultra-low latency connectivity to exchanges, consolidated data feeds, and electronic markets. This white paper describes the components and capabilities of CMDS - including support for over 100 international data feeds, consolidated order book capabilities, and high-availability features - and describes the competitive advantages of this groundbreaking platform.

## Overview of Confinity Market Data Distribution System

Confinity Market Data Distribution System (CMDS) is a product comprising integrated components for receiving, consolidating, and distributing market data to front-, middle-, and back-office applications within financial firms. This robust product offers flexibility, scalability, and reliability in virtually any customer environment, processing and delivering market data in a highly efficient manner.

CMDS provides connectivity to over 100 market data feeds, including a wide range of exchange data feeds, electronic communication networks (ECNs), and consolidated third-party feeds. Feed handlers receive, normalize, cache, manage, and integrate real-time market data for routing to subscribing clients. CMDS offers the ability to combine multiple order book feeds into a single consolidated view of the market. An entitlement system offers controlled access to market data. Enhanced administrative capabilities and built-in SNMP servers provide a wide range of operational management options, including real-time notification of performance issues or other critical events. A resolution service offers single namespace capability to shield applications from having to know the source of a particular instrument.

### Confinity Solutions Market Data Platform





High-performance, flexible, scalable middleware provides connectivity between data sources and subscribers, with intelligent message routing and APIs for publishing and consuming data. The messaging middleware can be configured at publishing, distribution, and client access tiers to use reliable multicast or point-to-point transports to provide scalable integration between applications.

CMDS can be deployed over a wide range of physical network designs. Object distributor components can dynamically discover each other and communicate by way of highly efficient and reliable connections, manage services, and route requests and responses.

The CMDS distributed architecture is designed for high availability, so no single component failure can cause an interruption in service. Requests are shared equally among all participating sources of a service, allowing load balancing across multiple sources of data. The client connection components are used to selectively route requests, responses, and update traffic bidirectionally between applications.

API programming interfaces are available for C++, Java™, and Microsoft® .NET, and Microsoft Excel connectivity is also available. A development toolkit provides the components needed to help develop connections between CMDS and customer applications, and provides the ability to build custom feed handlers.

The sections that follow describe the main system components included in the CMDS product.

### Feed handlers

Feed handlers are available for American, European, and Asian direct exchange feeds, ECNs, and consolidated feeds. Level 1 and Level 2 feed handlers receive incoming data; normalize, cache, and manage it; and integrate value-added content into the incoming data stream. Level 2 feed handlers maintain full- or partial-depth book information. Other feed handlers support consolidated feeds such as Bloomberg B-Pipe, B-Pipe On Demand, and IDC PlusFeed.

### Consolidated book

CMDS lets you see the full market depth by merging order books from several major liquidity pools. The consolidated book capability aggregates data from feeds of the same currency and overlapping instruments. For example, the following North American Level 2 feeds can be processed in any combination:

- NASDAQ TotalView
- NYSE ARCABook
- NYSE OpenBook Ultra
- BATS PITCH

Data can be aggregated across all feeds, by market maker, market center or price level. You can view order books from each independent market center, or combine any number of order book feeds into a single consolidated view. A two-tiered consolidated order book is available to handle increased feed throughput and to support multiple book managers over a range of instruments.



## Confinity Market Data Distribution System Feed Handler Coverage



### Platform services

Additional services provided as part of the platform include the permission service, resolution service, distributed caching service (with delay adapter and conflation adapters), and SNMP monitoring enablement components.

- Permission service—The permission service manages user entitlements to the market data instruments in the CMDS environment. It includes a permissions manager GUI and can use a MySQL database or any other SQL database. The database maintains an array of user statistics that can be queried with third- party tools or raw SQL queries to fulfill requirements for entitlement audit reporting.
- Resolution service—The resolution service resolves diverse feed sources entering the environment to a single name space for clients, eliminating the need for a subscriber to specify the data source. This service determines the appropriate source for fastest information retrieval. Through this service, user subscription or snap requests are resolved by configurable name and source resolution rules.



- **Distributed caching**—Distributed caching of feed data across a WAN provides a remote site with efficient access to market data. Additionally, information and samples are provided for development of a cache server adapter that can modify and broaden the server capabilities for customized use. Two examples are 1) a delay adapter that delays incoming data for a specified amount of time before forwarding data and 2) a conflation adapter that reduces the message rate of a subscription by combining multiple incoming updates into fewer outgoing updates.
- **SNMP monitoring enablement**—SNMP statistics are published for each enabled component. These built-in SNMP agents can be browsed with a third-party SNMP monitoring application, enabling integration with CMDS components. This allows you to optimize component operation and respond to system performance issues that are detected.
- **Administrative desktop**—The administrative desktop displays operating statistics of CMDS components. This is an Eclipse-based component available on all platforms. The permission service manager, a GUI to manage access control, is also available in this component.
- **Market data viewer**—The market data viewer allows you to view Level 1 and Level 2 data published from feed handlers in real time.
- **Subscriber and contributor**—The subscriber utilities are used to subscribe to market data from feed handlers, whereas the contributor utility is used to send application commands and data to feed handlers. Both subscriber and contributor are console applications that take input from standard input and send output to standard output. These features allow the utilities to easily be used in a scripted environment.
- **Microsoft Excel Add-in**—This component allows users to integrate market data delivered by CMDS feed handlers into Excel spreadsheets. The Excel add-in lets the user receive real-time updates, make snapshot requests, and contribute data back into the infrastructure to help support applications such as automated trading and risk management.

### Client connection components

These components include the object distributor, administrative desktop, runtime libraries, and subscriber and publisher utilities. They enable applications and authorized users to connect to the system. Object distributors, feed handlers and other registered components are monitored through the administrative desktop.

- **Object distributor**—The object distributor is the client/server connection point for the feed handlers, services, and clients in the CMDS environment. Major functionality in the object distributor includes tracking individual client publish and subscribe requests (request management), connecting object streams to client applications, determining the best source to fulfill a client request (service request routing), and the self-managing ability to form object distributor collectives. The object distributor automatically accepts and maintains connections to other object distributors for load balancing and failover assurance.

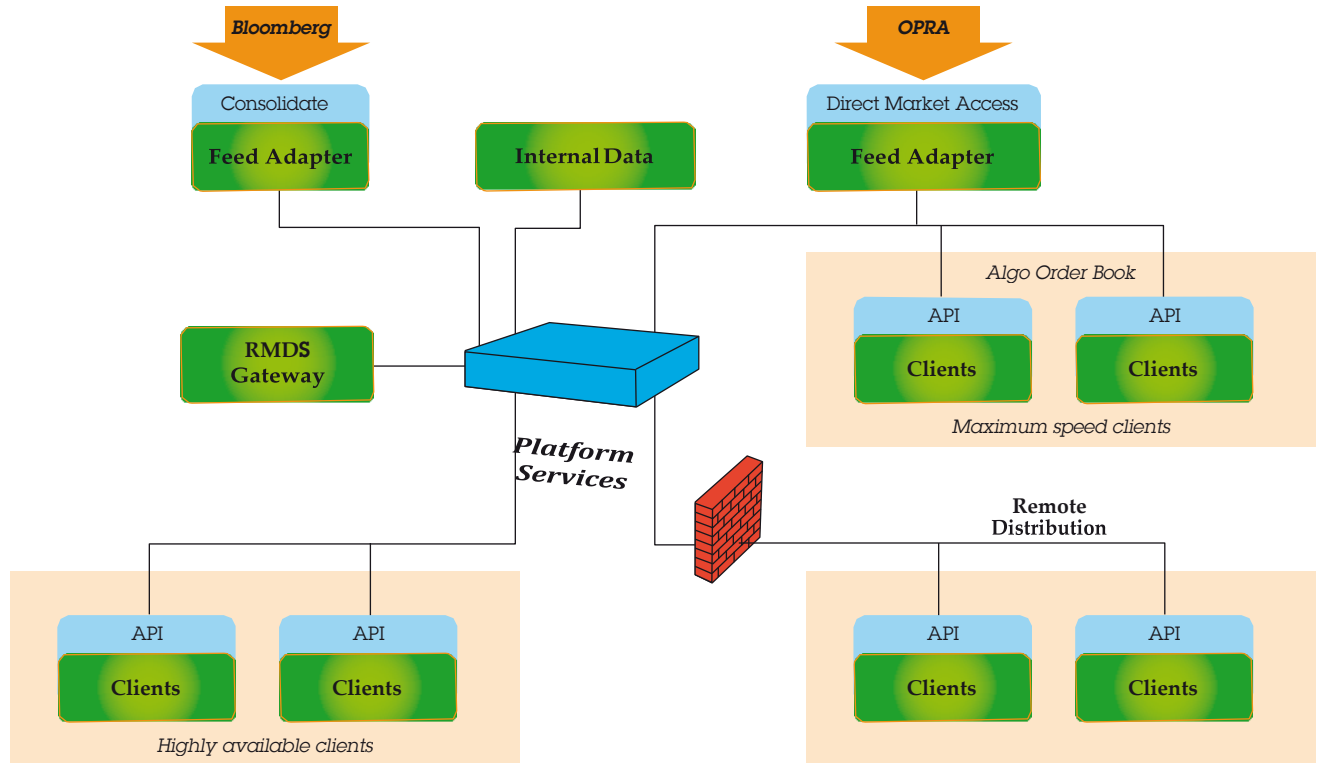
### Development Toolkit

The Development Toolkit includes the components needed to enable delivery of market data to client applications and to develop custom feed handlers. The toolkit includes all the elements of the client connection package, plus all necessary header files, reference information, and the Software Developer Kit for Java. It also includes a sample feed handler, a runtime package, full documentation, and all items necessary for Level 1 and Level 2 feed handler creation and operation in the CMDS environment, as well as support for the CMDS streamlined market data API.





## Confinity Market Data Distribution System Architecture



### General architecture

The modular design of CMDS is the key to its high performance and deployment flexibility. An object-oriented class library provides a common interface for all components.

### Ultra-low latency

Because speed and throughput are crucial, IT is increasingly pressured to deliver more messages per second to local and remote points. CMDS helps IT teams make speed a priority. Through integration with Confinity Low Latency Messaging – a messaging transport





engineered for the rigorous latency and throughput requirements typical of today's financial trading environments—CMDS provides high-speed multicast and unicast messaging support.

### High throughput

Confinity Low Latency Messaging is capable of delivering more than 45 million messages per second, and also provides support for native InfiniBand and for RDMA (Remote Direct Memory Access) over Ethernet, with latency measured as low as five microseconds for small message sizes. These next-generation interconnect standards offer high transmission rates and scalability to enable higher throughput with even lower latency, reduced latency variability, and low CPU consumption. CMDS supports both multicast one-to-many distribution and point-to-point (unicast) messaging capabilities to enable market data delivery to local or remote clients.

### Modular design

Redundant CMDS feed handlers interpret feed formats and build a cache of objects that are published only when requested, and then sent to the object distributor for delivery. Multiple object distributors can be deployed throughout a LAN, WAN, or Internet enterprise environment. Self-managing object distributors communicate with each other to form a collective middleware layer directing traffic and routing requests from subscribing clients. This architecture can be cascaded to form a two- or three-layer fan-out for high data volumes or clients. A large deployment typically contains redundant object distributor and service components. The self-regulating design of the object distributors allows for load balancing, source stream failover recovery, automatic rerouting around a failed object distributor, and sharing of available client services.

A distribution tier can be implemented using either the multicast or unicast capability provided by the embedded Confinity Low Latency Messaging. In this scenario, the object distributor is used to establish the initial connection. Once the connection is established, data is published directly from the feed handlers to reliable multicast or unicast channels, bypassing the object distributor to minimize latency. Multicast channels are configurable and load balanced to allow for partitioned distribution of multicast services.

Client applications can be written in C++, Java or .NET, and connect to the object distributor through a common library. If the connection is lost to an object distributor, a new connection to another object distributor will be made by the library. Services and data may be interrupted and then reestablished; client applications may then issue requests and receive replies and updates as before.

Basic operation of CMDS can be delivered with just a feed handler, object distributor, common libraries, and client receiver. Functional modules such as the resolution or permissions service can be added to the deployment when needed. Object distributors are designed to accept other services communicating with the class library in addition to the basic system.

### High availability

CMDS is designed for high availability, with a distributed architecture employing redundant components at all levels. No single component failure can cause an interruption in service, and the system can be configured such that even multiple component failures do not cause interruptions of service.



## Deployment flexibility

A platform deployment can be overlaid on virtually any physical network design and is independent of location. Middleware components can be added dynamically to the system and automatically become part of the object distributor collective. The middleware extends natively throughout a LAN. The distributed cache server offers the ability to cache data remotely to help support distributed deployment across WAN environments. Data and services can be sent from a central site over a point-to-point connection to a remote LAN, where information can be distributed to many clients using a reliable, high-speed, multicast transport. Snapshot requests can be satisfied locally, which can dramatically reduce network traffic between sites.

## In depth: Product components

The following are in-depth descriptions of the components that make up CMDS.

### Feed handlers

CMDS feed handlers are based on the class library and process Level 1, Level 2, and consolidated feeds. Consolidated book functionality is also available. Feed handlers receive, normalize, cache, manage, and integrate real-time market data sourced directly from exchange feeds and ECNs. They provide access to quote, trade, and full depth and breadth of all feed vendor information, including global exchange data, global depth of book, and fields of information published for symbols for many national and international feeds.

Feed handlers are designed to accommodate rapid integration of additional data sources. Feed handler functionality includes:

- Configurable multithreaded implementation.
- Automated data management features including composite records, participant records, market maker records, option chaining, option expiration, intra- and interday field processing, and static and fundamental data integration.
- Support for publish and subscribe distribution by name and query.
- Support for data contribution by client applications.
- Feed processing distribution using splitting.
- Several order book views including aggregation by price or market maker.
- Data integration support for legacy systems include symbology, option chain records, forms, field ID assignments, field values, and message formats.
- Support for redundancy, load balancing, fault detection, failover, and recovery.
- Single point of management, monitoring, and control through the administrative desktop or SNMP MIBs and traps.

Specific feed handler capabilities include:

- Data normalization—Normalization of the incoming data stream helps ensure consistent object interfaces throughout the enterprise. Data normalization objects are records and grids; all data is self-describing through metadata.
- Value-added data—Real-time calculations made from message events in a direct exchange feed can be added to the feed as new fields in the message. Additions include high, low, open and close price, accumulated volume, up/down tick, net change, percent change, block volumes and counts, VWAP, and others.



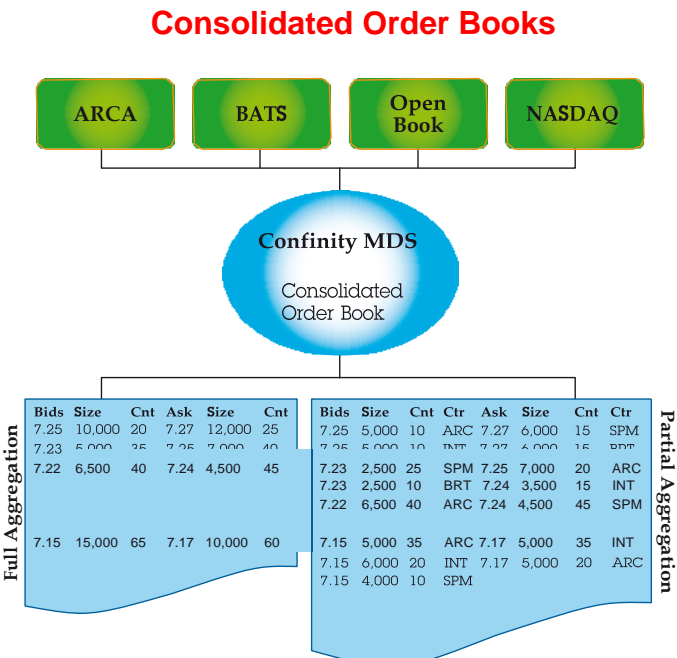
- Data blending—Customers can blend direct exchange data with sources of fundamental and reference data, including information from vendor data feeds. This combined data is presented as a single message. The tools and utilities provided help customers easily contribute or publish their own sources of static or dynamic data within the system.
- Custom data integration—Several methods for customers to contribute their own computed information to the system are available. Using the callout API within the feed handlers, customers can add data elements inline as data market feed messages are being received and processed, prior to publication for subscribing clients.

Some specialty feed handlers also provide:

- Dynamic data model based on vendor-specific metadata.
- Data translation through configuration.
- Symbology aliasing based on rules per asset class.

Consolidated order books

Consolidated book functionality is provided in a separate component within the Order Book feed handler. CMDS consolidated book functionality provides advanced data models to represent full depth of book information. Top-n price levels can be integrated into a traditional record-based data model and chains to accommodate older market data systems. CMDS lets you see the full market by merging order books from each of the major liquidity pools. Order books can be viewed from each independent market center, or any number of order book feeds can be combined into single consolidated views. Data from feeds such as ARCA, NASDAQ, OpenBook, DirectEdge, BATS, and other order book feeds is consolidated by aggregated market size at each price, or partial aggregation showing aggregate size at price in each market center, consolidation without aggregation, or other views.



The two-tiered consolidated book gives downstream clients a broad picture of the aggregated order data across all supported trading venues.

Object distributor

The object distributor (ITOD) is the middleware routing component for CMDS that supports request and response, and publish and subscribe functionality. The object distributor connects, services, and resolves object routing for all client and server applications. For availability and scalability, multiple object distributors can be deployed throughout a LAN, WAN, intranet, extranet or Internet environment.



The object distributor can also function as an enterprise gateway to other vendor environments. The object distributor reflects services, manages request and response traffic, and routes traffic between these enterprises in real time. It serves as an on-site integration component to extend shared or dedicated managed services from a common service facility.

Client/server applications participate in the environment through a session with an object distributor based on TCP/IP. Every object distributor supports multiple simultaneous client and/or server sessions; multiple object distributors comprise a collective in an enterprise system. Object distributors may be added to the system at any time.

### Resolution service

The sole function of the resolution service is to resolve client symbol requests to specific feed handlers. The resolution service does not interfere in the flow of response or update traffic within the system. Multiple name spaces can be accommodated to resolve requests for various feed handlers with the system.

The resolution service employs a single name space function to combine all feeds entering the system into a single virtual service that shields users from having to identify the source of a particular instrument. Single name space implementation does not introduce additional latency or overhead.

### Permission service

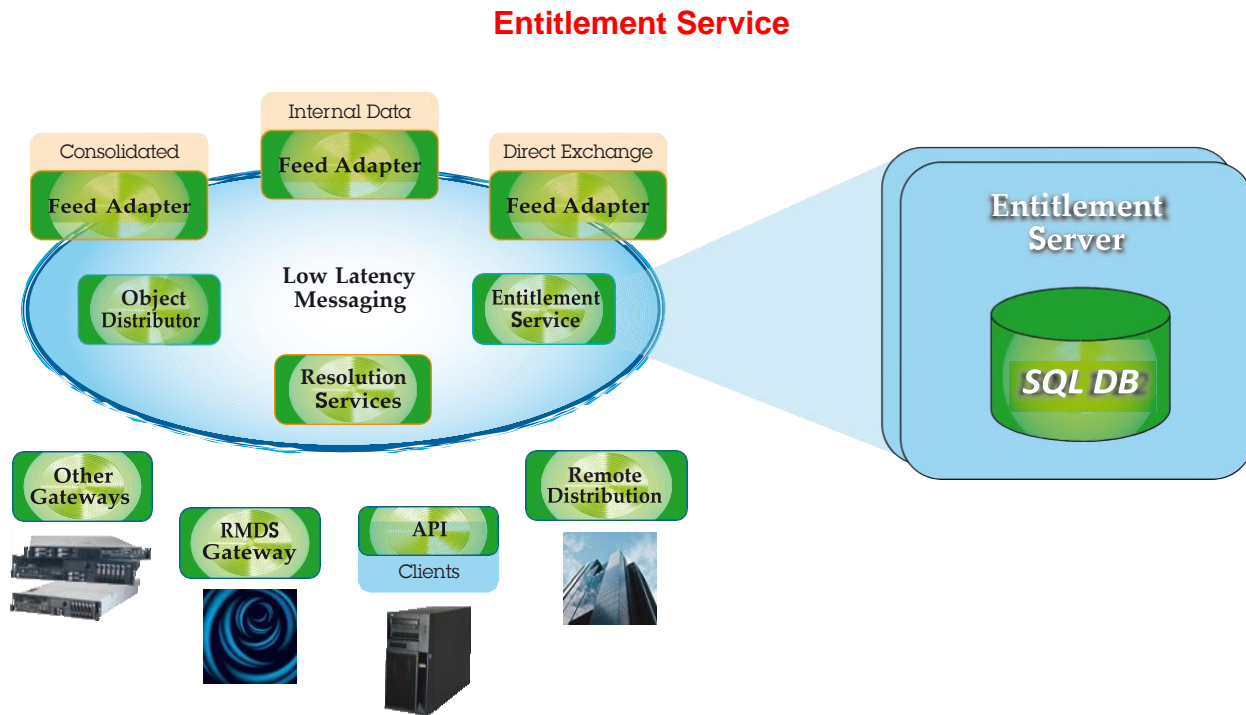
The permission service works with the bundled database to resolve entitlements to user requests for specific data and services. The service employs industry-standard user request

validation SQL queries. After the permission service successfully connects to the preconfigured database, it makes a connection to an object distributor.

CMDS offers a fully featured permission- entitlement service to authenticate users and control data access. Each time a user connects to the system, the user is authenticated based on the permissions granted. The system ensures only one connection per user and can disable a user, even during a session. Each symbol subscription is checked against the entitlements that have been established for that user.

CMDS provides two entitlement types— name based and content based. The basic entitlement type is referred to as name based or symbol based. The name-based entitlement is granted or refused based on matching service name, source name, and item or symbol name specified in the data item request against those specified in entitlement definitions assigned to a specific user. The content-based entitlement type is an extension to name-based entitlement. In addition to matching service, source, and item or symbol name, this entitlement will attempt to match a permission code string, sometimes called an attribute, returned by the feed handler when requested by the object distributor as a part of the entitlement resolution.

The permission manager is an Eclipse-based application that is integrated with the administrative desktop. It enables an administrator to add, delete, and modify user accounts, define entitlement rules, and associate entitlement rules to a user.



## Distributed caching

Distributed caching provides a remote site with a last-value cache for local access. Distributed caching, including a delay adapter and conflation adapter, is a function of the cache server.

The distributed caching service employs the cache server to efficiently provide market data access across a WAN connection to a remote LAN. The cache server can be connected to one or more remote object distributors to provide the following functionality:

- Subscription to real-time market data
- Cache services
- Symbol translation

The conflation adapter function allows the cache server to be configured to reduce the update rate of a subscription. This is accomplished by combining multiple incoming events into fewer outgoing events. While individual events may be suppressed, downstream clients will always be forwarded the current value of the cache. Data conflation can significantly reduce network bandwidth requirements and data consumption by subscribed applications.

The advantages of employing distributed caching include:

- Statistics generation and storage
- Optimized logging
- Proxy subscription to other CMDS services
- Seamless failover to a backup server

A delay adapter function allows the cache server to be configured to delay subscription updates by a predefined amount of time. Delaying the distribution of market feeds is a common scenario for enabling the use of different pricing structures. In this way, CMDS can be configured so that data will not be released before the desired amount of time.

Real-time monitoring and logging capabilities help provide an audit trail that demonstrates that the data was delayed appropriately.

- Offloading feed handler transmissions.
- Network connection and bandwidth preservation.
- Service consolidation.
- Failover and recovery.

## Administrative desktop

The administrative desktop (ITAD) is an Eclipse-based GUI application that manages components through its connection with the object distributor. The administrative desktop interface includes menu selections for system monitoring, application configuration, command and control.





Market data viewer

The market data viewer enables viewing of updates to specified symbols as they are published. The market data viewer is an Eclipse-based GUI application that displays real-time published updates to symbols chosen in a Watchlist. The market data viewer displays both Level 1 and Level 2 data with statistics.

Microsoft Excel Add-in

The Microsoft Excel Add-in allows market data to be requested from the CMDS environment into a Microsoft

Excel spreadsheet for further analysis.

The add-in only needs to be added to Excel once for continued access to the functions. Samples of Microsoft Excel spreadsheets are provided as a starting point for retrieving real-time data. The spreadsheets contain wrapper functions that simplify the Excel add-in interface.

Development tools

The CMDS Development Toolkit supports three programming styles with an object-oriented class library, a market data message-level interface, and a callout API,

which supports the loading of client application code into the feed handler process. The CMDS Development Toolkit comes with additional components needed for application development such as header files. The .NET version of the class library exposes the C++ classes through the market data API (IMDA) and makes them available to other programming languages.

Class library

The CMDS class library (ITCL) is an object-oriented class library that acts as a common interface for real-time client/server communications. The class library also includes utilities used by the CMDS components.

Description	Value	Time
Feed Up State	1	02/11/10 18:02:02
Messages Processed (previous second)	30	04/21/10 16:28:11
Messages Processed (this hour)	298429	04/21/10 16:28:11
Messages Processed (previous hour)	15022506	04/21/10 16:00:00
Messages Processed (today)	95669379	04/21/10 16:28:11
Messages Processed (yesterday)	80246878	04/21/10 00:00:01
Filtered Messages (today)	5088	04/21/10 02:45:02
Filtered Messages (yesterday)	37398230	04/21/10 00:00:01
Retransmitted Messages (today)	0	03/02/10 00:00:00
Retransmitted Messages (yesterday)	0	03/03/10 00:00:01
Internal Message Errors (today)	0	02/11/10 18:01:52
Internal Message Errors (yesterday)	0	02/11/10 18:01:52
Message Import Errors (today)	0	02/11/10 18:01:52
Message Import Errors (yesterday)	0	02/11/10 18:01:52
Sequence Numbers Missed (today)	413	04/21/10 09:54:40
Sequence Numbers Missed (yesterday)	846	04/21/10 00:00:01
Sequence Numbers Repeated (today)	0	02/11/10 18:01:52
Sequence Numbers Repeated (yesterday)	0	02/11/10 18:01:52
Sequence Number Gaps (today)	7	04/21/10 09:54:40
Sequence Number Gaps (yesterday)	19	04/21/10 00:00:01
KBytes Processed (previous second)	2	04/21/10 16:28:10
KBytes Processed (this minute)	63	04/21/10 16:28:10
KBytes Processed (previous minute)	487	04/21/10 16:28:01
KBytes Processed (this hour)	34425	04/21/10 16:28:10
KBytes Processed (previous hour)	1100045	04/21/10 16:00:00
KBytes Processed (today)	7530736	04/21/10 16:28:10
KBytes Processed (yesterday)	8806362	04/21/10 00:00:01
Line state action flags	3	02/11/10 18:01:52



The class library interface is an abstract object model that includes convenient wrappers for:

- Session management.
- Service provisioning.
- Object caching.
- Data encapsulation.
- Threading.
- Self-describing objects.
- Custom object definitions.

Within the class library, an abstract network interface model supports configurable adapters. This allows client/server applications based on the class library to use different network distribution environments using the CMDS object distributor. This class library is ideal for the development of vendor-neutral interfaces to client/server applications handling real-time market data.

### Market data API

The CMDS market data API is designed to publish and subscribe to market data events—rather than objects, as the class library does. Implemented on the same foundation as the class library, the market data API accommodates applications that are interactive processing events associated with market data, such as trades, quotes, cancels, corrections, summaries, add order, delete order, and so on. The market data API hides the published details of determining these events, and provides a convenient way to implement an event-based application. The market data API supports Linux®, Microsoft Windows Server 2014 and 2016 and Windows® 7 and 10.

Market data API programming interfaces are available for C++, C#, VB, Java, and .NET. The C++ version of the market data API includes a DLL for Windows as a runtime component and a shared object library for Solaris and Linux. The CMDS Development Toolkit comes with additional components needed for application development for Linux, Windows, AIX, and Solaris.

The .NET version exposes the C++ classes through .NET and makes them available to other programming languages such as C#. .NET is an additional DLL available for Windows only and also uses the class library DLL. The Java version exposes the C++ classes through JNI™ to the native Java machine.

The Java API utilizes the market data API available for Linux, Solaris, and Windows to provide a high-performing Java capability without sacrificing typical platform availability.

### Callout API

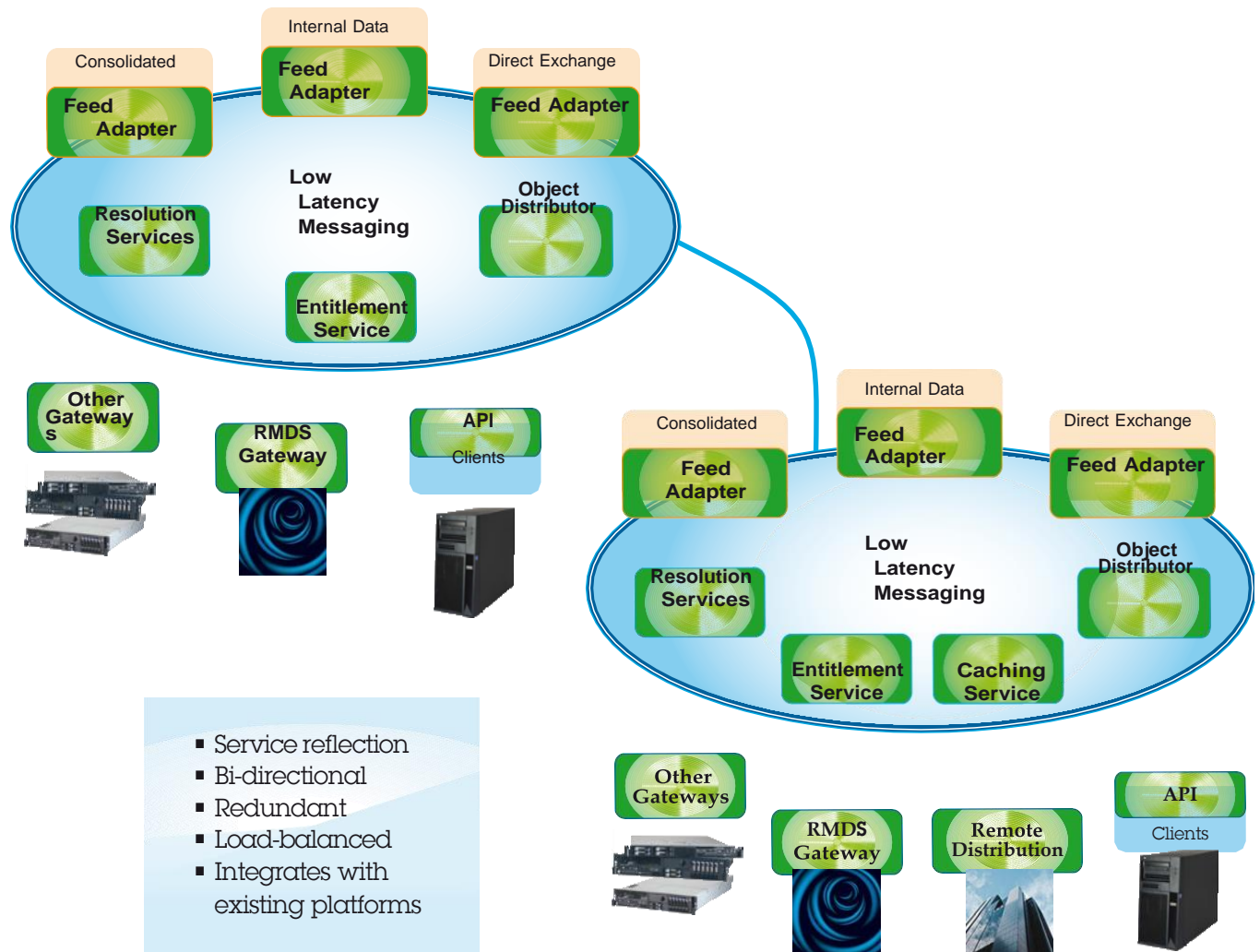
Unlike the previously described APIs, which are interactive and utilized in conjunction with the CMDS platform, the C++ dynamic callout API is embedded with the feed handler. The loaded shared library becomes part of the feed handler process and allows access to the market data after it has been normalized, but before it has been published to the system. Using this API is the fastest and most direct way to obtain market data. Typical uses of this API include:

- Data modification—Add to or modify data within the object, such as including value-added calculations or other information for publication to the enterprise.
- Immediate access—Use the normalized data directly for application processing.
- Custom integration—Retrieve the data and serialize it into a custom format for input to any application. Leverage the feed handler processing functionality then customize the data integration into another system.





## Global Market Data Distribution LAN/WAN Connectivity





Like the market data API, the callout API is available for Solaris, Linux, AIX, and Windows operating systems. Additional components needed for application development are available in the CMDS Development Toolkit.

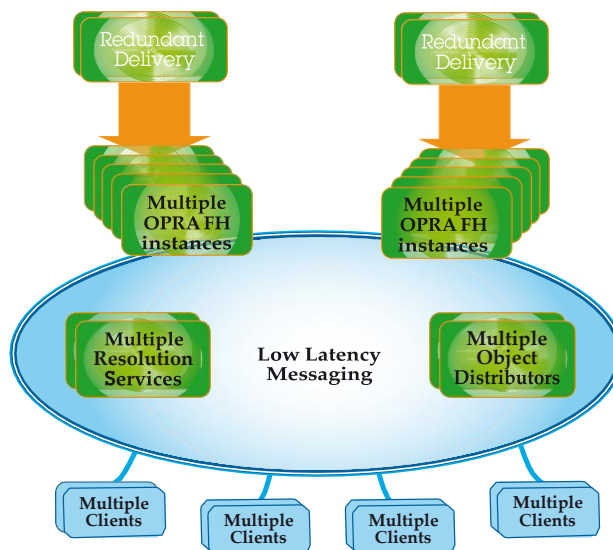
### Flexibility and scalability

To meet the performance and integration demands of market data users, the CMDS architecture offers a highly flexible and easily scalable set of feed handlers, object distributors, and application programming interfaces. Since all components are multithreaded, they are capable of taking full advantage of multiprocessor hardware architectures and operating systems. Throughout the system, scaling takes place independently and dynamically by simply adding additional component instances while the system is running. This is also useful when upgrading a system since new releases are designed to be backwards compatible, allowing previous individual instances to be taken down and current releases to be brought back up dynamically.

Service scaling allows users to distribute processing for individual feeds horizontally across any number of physical servers and to run multiple feed handler instances within each server, fully utilizing all available processing cores while preserving complete system functionality. More than simple load

balancing, which most vendor platforms support, service scaling makes it possible to assign data feed processing to an array of servers as necessary to keep pace with growing message rates. The associated middleware intelligently resolves requests on behalf of client applications and nothing is lost relative to quality of service. Most importantly, there is zero added latency.

### Vertical and Horizontal Scaling





## Ultra-low latency, high throughput messaging transport

CMDS is integrated with Confinity Low Latency Messaging (CLLM) to support extremely low (microsecond) latency and high message volumes (ranging from many thousands to millions of messages per second). Confinity Low Latency Messaging is a highly optimized messaging transport that facilitates the high-speed delivery of market, trade, reference and event data in front-office operations as well as between front-, middle- and back-office operations. It achieves breakthrough speed by efficiently packing data and exploiting the IP multicast infrastructure in a daemonless fashion to eliminate network connections.

Confinity Low Latency Messaging offers the following specific features to support high-speed message delivery.

- Multiple messaging transports—In addition to the reliable multicast transport, Confinity Low Latency Messaging offers lightweight, point-to-point UDP transport, and reliable, point-to-point, unicast messaging over TCP/IP.
- Traffic control features—Confinity Low Latency Messaging adds traffic control features on top of the transport layer to avoid competition between receivers and between streams with subsequent delays and data loss.
- Highly available message transmission—All Confinity Low Latency Messaging transports enable high-availability distribution by implementing numerous stream failover policies to allow seamless migration of message transmission.

- Native InfiniBand—Support for Native InfiniBand, the next-generation interconnect standard that offers high transmission rates and scalability, enables higher throughput with even lower latency, reduced latency variability, and low CPU consumption.
- RDMA over Ethernet—Support for Native RDMA over Ethernet enables a network adapter to transfer data directly from one computer's memory to another, bypassing the operating system, resulting in reduced CPU consumption.

## Compatibility with existing systems

Many organizations have existing market data systems supporting a large number and variety of applications. Through InfoDyne technology, CMDS provides out-of-the-box integration with RMDS platforms.

ITOD/Gateways are used to selectively route requests, responses, and update traffic bidirectionally between CMDS and these other systems.

CMDS can be integrated with multiple other systems simultaneously, ensuring a low cost of deployment and ownership, since feeds and feed handlers do not need to be deployed multiple times per system. This provides tremendous flexibility when servicing multiple business units from the same market data platform and avoids the need for a “big bang” integration effort.



### Designed for mission-critical operations

CMDS is designed for continuous operation as a mission-critical system that needs to be available 24x7.

Site failover strategies can be employed during software or hardware installations such that client service never needs to be interrupted.

### System monitoring

CMDS includes a combination of ITCL- based administrative components and SNMP utilities that provide an extensive range of operational management capabilities. These capabilities include interoperable subscription and contribution utilities, administrative agents, administrative desktop applications, standard SNMP publication, and more.

### Latency monitoring

CMDS feed handlers monitor latency continuously on every receiving port to enable customers to detect latency and to ensure that data vendors and extranet providers are satisfying their SLA commitments.

### Administrative desktop

The administrative desktop (ITAD) is an Eclipse-based GUI application with an easy-to-use interface that lets a system administrator view the entire system and drill down to view specific statistics for an individual component. All CMDS components publish extensive status information and statistics, which can be monitored through a common point of access using the administrative desktop.

### SNMP integration

Every metric in the system has an associated SNMP MIB ID that is used by the SNMP agents to publish these metrics. Standard SNMP utilities, such as IBM Tivoli® and HP OpenView software, use the MIBs to retrieve these metrics.

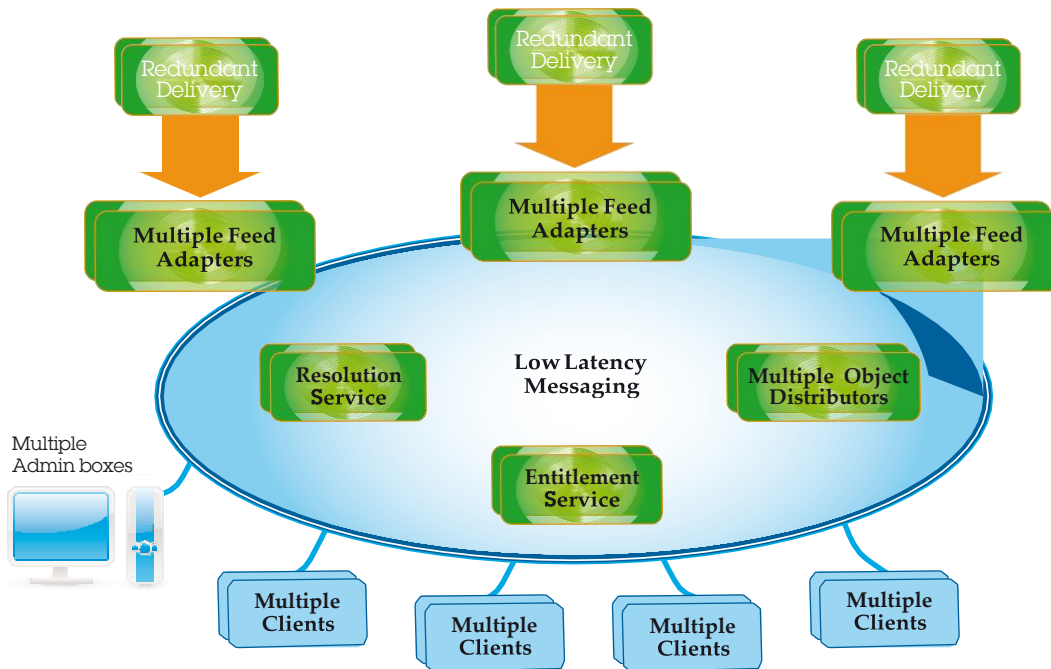
Through the administrative agents, the administrative desktop, and SNMP MIBS, operators can be notified in real time of any interruption of feed service, infrastructure disconnections, component failures, capacity warning, latency breaches, missed messages, and a host of other critical events that may occur in the system. Confinity Solutions can also assist customers by providing component execution and monitoring scripts that report problems through email, SMS, and other mechanisms.

The screenshot displays the administrative desktop interface. On the left, a tree view shows the system hierarchy: CTA 1A, LOG, STATS, Feed: CTA, Channel: CQS1, Port: CQS11, Day State[CQS11], Delta[CQS11], Port Agent: METROTECH\_C, Port Agent: WATERSTREET, SequenceNumber[CQS11], Test State[CQS11], Transmission State[CQS11], Port: CQS110, Port: CQS111, Port: CQS112, Port: CQS113, Port: CQS114, and Port: CQS115. On the right, a table titled 'Statistics' shows various metrics and their values over time.

Description	Value	Time
Delta update interval (sec)	5	04/19/10 13:47:58
Current delta (ms)	2	04/19/10 14:02:05
Current delta	00:00:00.002	04/19/10 14:02:05
Avg delta (ms) (this minute)	2	04/19/10 14:02:05
Max delta (ms) (this minute)	2	04/19/10 14:02:05
Min delta (ms) (this minute)	2	04/19/10 14:02:05
Avg delta (ms) (previous minute)	8	04/19/10 14:02:00
Max delta (ms) (previous minute)	11	04/19/10 14:02:00
Min delta (ms) (previous minute)	3	04/19/10 14:02:00
Avg delta (ms) (this hour)	9	04/19/10 14:01:47
Max delta (ms) (this hour)	-11	04/19/10 14:00:17
Min delta (ms) (this hour)	2	04/19/10 14:02:05
Avg delta (ms) (previous hour)	17	04/19/10 14:00:01
Max delta (ms) (previous hour)	1095	04/19/10 14:00:01
Min delta (ms) (previous hour)	2	04/19/10 14:00:01
Avg delta (ms) (today)	16	04/19/10 14:00:47
Max delta (ms) (today)	1095	04/19/10 13:47:59
Min delta (ms) (today)	2	04/19/10 13:50:59
Avg delta (ms) (yesterday)	0	04/19/10 13:47:58
Max delta (ms) (yesterday)	0	04/19/10 13:47:58
Min delta (ms) (yesterday)	0	04/19/10 13:47:58



## Confinity Market Data Distribution System High Availability



### High availability

CMDS components can be configured for high availability through redundancy, thus eliminating any single point of failure in the data path. Data-consuming applications can be configured to have a primary and an alternate object distributor session, and object distributors have access to all feed handler services. Feed handlers are deployed in

redundant pairs, and each feed handler processes two copies of a data feed, which are delivered through different network paths. CMDS failover algorithms detect and respond automatically to quickly isolate single component failures to provide high levels of service availability.



### Advantages of CMDS

Some of the major advantages of using CMDS include:

- A market data source-neutral alternative for delivery of market data that can address challenges such as increasing market data and trade volumes, rises in algorithmic trading, market fragmentation, and regulatory changes.
- Feed handlers that can handle streaming data coming into or out of the CMDS infrastructure and provide connectivity, data processing, and control and administration capabilities.
- Support for over 100 data feeds, including major U.S. and international data sources, with connectivity to exchanges, ECNs, and consolidated data providers.
- Consolidated order book capability, providing the ability to combine any number of order book feeds into a single consolidated view.
- Low data latency and high throughput on an integrated, high-performance, high-availability platform, with support for high-speed multicast and point-to-point message transports.
- Distributed cache server with optional delay and conflation adapter functionality to support a range of enterprise distribution requirements.
- Easy integration of direct and aggregated data feeds into applications through a flexible interface for front-, middle- and back-office connectivity.
- A manageable platform supporting entitlements, performance monitoring, single namespace, and an integrated Eclipse-based administrative desktop.
- Microsoft Excel connectivity to export data for analysis.
- Full support, including services and consulting.
- A flexible pricing model that provides an attractive total cost of ownership.



### For more information

To learn more about Confinity Market Data Distribution System please visit our website at [www.confinitly-solutions.com](http://www.confinitly-solutions.com) or contact us at [info@confinitly-solutions.com](mailto:info@confinitly-solutions.com)



Confinity Solutions GmbH  
Mergenthalerallee 45-47  
D-65760 Eschborn,  
Germany

© Copyright Confinity Solutions 2016, 2017  
© Copyright IBM Corporation 2010-2015

Produced in Germany June 2017  
All Rights Reserved

IBM, the IBM logo, ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other

Java and all Java-based trademarks and logos are trademarks of Oracle Corp. or Sun Microsystems, Inc. in the United States, other countries, or both.

Linux is a trademark of Linus Torvalds in the United States, other countries or both.

Microsoft and Windows are trademarks of Microsoft Corporation in the United States, other countries or both.

Other company, product or service names may be trademarks or service marks of others.





