TWINOAKS COMPUTING INC.
PRACTICAL MIDDLEWARE EXPERTISE

2012

What can DDS do for Android?

Twin Oaks Computing, Inc

755 Maleta Ln, Suite 203
Castle Rock, CO 80108
720-733-7906
855-671-8754 (toll free)
www.twinoakscomputing.com

Contents

Abstract

Today's Android developers typically build their applications projects without middleware. This is understandable considering most early apps did not communicate off the Android device. However, with Communications Middleware, this is changing.

The number of activated Android devices continues to grow, with Android holding a 43% share of the US mobile market in 2011 (that's almost 50 million active users).[1] *Along with this popularity, software developers in both the commercial and DoD industries are finding new and valuable uses for more complex and distributed apps on these mobile, handheld devices. In addition, many project managers would like to make their existing software Android compatible.*

Communications Middleware like CoreDX DDS provides numerous benefits to distributed software systems, and these benefits can now be taken advantage of by Android apps.

This paper will give some background information on Communications Middleware, DDS, CoreDX DDS, and Interoperability, and how they apply to Android.

[1] http://www.theverge.com/2012/1/7/2689585/neilson-2011-media-numbers-tv-android

What is Communications Middleware?

Communications Middleware is computer software that enables two otherwise separate software components, processes, and/or applications to exchange information, either within one device, or between multiple devices. It is a specific kind of Middleware: the layer that lies between the operating system (Android, Linux, Windows, etc.) and system applications (accounting software, media players, office productivity suites, etc.), that allows for communications.

The purpose of Communications Middleware is to simplify the designing, programming, and managing of software applications by streamlining the way these applications receive and process data.

Communications Middleware is used in a wide variety of software systems, from mobile devices (Android phones, PDAs, Kindle Fires, iPads, etc.) to enterprise and database systems. The equipment in these systems varies in screen and visual display capabilities, bandwidth capacities, and processing power.

The bottom layer consists of two or more different devices, each using their own version of an operating system. When we add middleware, those different devices can now communicate and use the same application, even though they use different operating systems.

Communications Middleware can understand and support multiple programming languages (Java, C, C++, PHP, Ruby on Rails, etc.). We can use an Android phone and a PC here as an example. They both function in vastly different capacities, but with Communications Middleware are able to "talk" to and "work" with each other. This holds true for devices of similar capacities with different operating systems as well.

Why use Middleware?

A wide variety of operating systems are being used in today's software development efforts: Android, Windows, Linux, and QNX, just to name a few. These operating systems communicate data differently, just as different hardware types (cell phones, computers, printers, etc.) store and retrieve information in a variety of ways. With the increasing popularity of Android apps, many developers would like to make their existing applications Android compatible. This translates into an expensive problem when your project wants to exchange information between two diverse systems, potentially costing you and your business precious time, money, and resources.

The solution to this problem is DDS. Using a Communications Middleware such as CoreDX DDS reduces system complexity. While different Communication Middleware technologies provide different features and benefits, they all strive to provide application portability across different operating systems and hardware, reduce development cost, and simplify the resulting application code.

What is DDS?

Data Distribution Service (DDS) is a type of Communications Middleware whose concept was standardized and is currently managed by the Object Management Group (OMG). [2]

DDS simplifies software systems, and reduces risk and costs through development, integration, deployment, and lifetime maintenance of distributed software systems.

Historically, DDS has been used in large DoD systems to satisfy Open Architecture requirements for Extensibility, Maintainability, Composability, and Interoperability, but only in the larger computer components of these systems. Now, with the availability of small-footprint DDS implementations, many other applications can benefit from standardized publish subscribe communications, including Android apps.

The DDS Standard contains an easy to use, well defined Application Programming Interface (API). This allows the developer to write *portable* code, code that will work with any compliant DDS implementation. The DDS standard references the Real Time Publish Subscribe (RTPS) Wire Protocol standard which defines the wire protocol for DDS communications. This allows applications built with different DDS implementations to communicate, or *interoperate*, with each other. Users of DDS do not tie themselves to a particular vendor, but to a standard, and can change or intermix DDS vendors throughout the development and deployment cycles.

Each application communicating over DDS contains the DDS API and provides the discovery, and other required communication details. DDS simplifies communications processes among different system types, making distributed development easier, faster, and more reliable. A DDS Communications Middleware simplifies your Android project from development through initial deployment and maintenance over the life of the system.

How does DDS work?

DDS is in charge of transferring information: Information is transferred from publishers (producers and senders of messages) to subscribers (consumers and receivers of messages). Subscribers and publishers employing DDS can use different platforms or operating systems and still communicate with each other. Exchanges can take place through tens of thousands of devices at the same time, each one of which can be publishers, subscribers, or both simultaneously.

[2] http://www.omg.org

Systems that use DDS to communicate can do so independently of each other: They do not rely on each other's systems to send and process information. A publisher can still publish information even if there is no subscriber seeking the information. A subscriber can receive information from other publishers if the original publisher it was getting information from fails[3].

DDS automatically knows how to send and receive messages with other DDS users: By design DDS is able to conclude which users should receive messages, where these users are located, and what to do if the receiver is unavailable. This simplifies data distribution, lessens the code required to perform message delivery (and less code means more efficiency), and thus saves time.

DDS participants can be on the same machine or across a network: the application uses the same DDS API for communications. Because there is no need to know or configure IP addresses, or take into account the differences in machine architectures, adding an additional communication participant on any operating system or hardware platform becomes an easy, almost trivial, task.

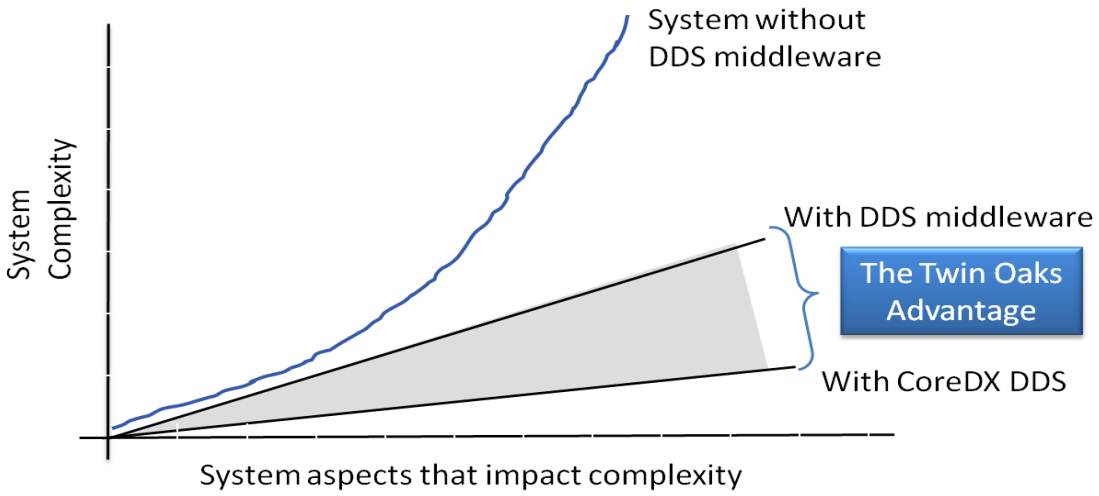
Each version of DDS can perform the same minimum set of functions in the same way with the same results. This is referred to as an "open standard system": system components from different manufacturers can be replaced and/or take over for each other with minimal or no changes to the larger systems in which they operate. This saves costs and avoids vendor lock-in.

DDS works in "real time": With very low overhead and efficient processing, messages are sent with minimal latencies (generally measured in the microseconds). It has a flexible architecture that is also *scalable*: it can adapt to processing both large and small amounts of data.

What are the Benefits of DDS?

DDS is widely adopted across a variety of industries, including some of the most mission-critical systems within the United States Department of Defense[4]. DDS is also being used in a growing number of commercial applications, including smart vehicle control, high-speed stock trading, consumer electronics, telecommunications, manufacturing, power generation, medical devices, and simulation. As Android use and popularity increases, we expect to see this trend among Android Apps as well.

[3] http://www.ucsarchitecture.org/downloads/DDS%20Exec%20Brief%20v20l-public.pdf
[4] http://www.ucsarchitecture.org/downloads/DDS%20Exec%20Brief%20v20l-public.pdf



DDS Reduces Risk:

DDS ensures consistency: Users of DDS can make changes to one system without the other system being adversely affected. Time is saved as less design time is allocated to determining how to get these systems to "talk" to each other.

*DDS automatically switches between publishers if the primary publisher fail*s. For example, once programmed, a publisher knows to "re-try" in 10 milliseconds, 10 minutes, every hour, or to drop the message all together, etc. if it is unable to reach a subscriber, and vice versa. In addition, subscribers always get the information that most closely matches their needs. If the information they seek is unavailable, they get the next best information. The system will automatically switch back to the information that most closely matches their needs when it becomes available.

DDS has no single point of failure: Systems that use DDS to communicate can do so independent of a server or service, and independently of each other. They do not rely on each other's systems to send and process information. A publisher can still publish information even if there is no subscriber seeking the information, or if a subscriber becomes "lost" for any reason. A subscriber can search for other publishers if the publisher it is getting information from fails or is lost.

DDS filters data for unique users: Each user only receives the information they need (or are intended) to receive. Consider online banking. The information is available to anyone who can access the web, as long as they have the correct username and password.

*DDS can be used wirelessly to communicate information***:** For example: handling secure transactions via Smartphones and financial institutions, or scanning and tracking systems for package delivery systems. DDS provides high performing, **reliable** communications over un-reliable wireless networks, including Wi-Fi, Bluetooth, and cellular networks.

DDS is reliable and always available - interactions with other services or application's are independent from network services, meaning they are always available for users (the server can't be "down" because of too many users, etc.) Data is cached by the publisher until all subscribers have received the information, so even if the network is unavailable the information is not lost. The publisher and subscriber merely try again.

DDs has the ability to tailor communication behavior – Quality of Service (QoS) policies allow the user to configure over 22 distinct items of communications behavior, providing fine-grained control to meet your communication requirements[5]. For example: reliability requirements, storage requirements, data presentation requirements, data filtering requirements, and redundancy or failover requirements (more than one path to communicate information).

DDS Reduces Cost:

DDS cuts development lifecycle cost: When disparate systems need to be integrated, instead of building a new system from the beginning, DDS can be deployed to facilitate communications and the project can continue. This saves both time and labor cost[6].

Administration and maintenance expenses are reduced with DDS: Standardized programming and communication interfaces simplify administration and maintenance of DDS-enabled systems. It is easy to replace a system component because the other components don't have to change; the new component is simply accepted. It is easy to remove a component because other components will continue to exchange information when the removed component is gone. It is easy to add a device: new publishers and subscribers can be added with no change to existing components.

[5] http://portals.omg.org/dds/category/keywords/qos
[6] http://www.ucsarchitecture.org/downloads/DDS%20Exec%20Brief%20v20l-public.pdf

CoreDX DDS

CoreDX DDS is a full-featured DDS implementation that comes in a surprisingly small package, perfect for Android devices. The entire "Android library", containing full interoperability with the RTPS wire protocol and support for all the standard, and a some additional QoS policies, is a mere **500 KB.** That's Kilobytes, not Megabytes! The CoreDX DDS Ping application (text-based application built to test interoperability over lots of QoS policies) is just 250 KB.

Platform Operating Systems and Transports

CoreDX DDS supports a wide variety of platforms, including:

- Android,
- ARM
- Bluetooth Network
- Cellular Network
- Linux
- Lynx OS
- NexusWare
- QNX
- Solaris 10
- VxWorks
- WI-FI Network
- Windows

Small Runtime Footprints

CoreDX DDS truly has a small footprint. Small from every angle, the full-featured CoreDX DDS library is measured in KiloBytes not MegaBytes. This compact implementation is truly unique in the middleware industry, and allows CoreDX DDS to benefit a wide range of Android devices. CoreDX DDS is small from every angle: small library size, low line of code count, and minimal run-time resource requirements. CoreDX DDS has been deployed on a single CPU Intel Pentium system with just 640K RAM.

Low Line of Code Count

Every line of code in a software product has a cost associated with it, and not just the cost of certification for safety critical applications. Each line of code has the potential for increasing the number of instructions that must be executed, and degrading the overall performance of communications. In addition, each line of code has the potential for a programming error, or bug, and more lines of code make it difficult to track down and identify such errors.

These are fundamental concepts in any software development project, and truly experienced software engineers understand the importance of writing code that is well thought out and compact. With CoreDX DDS, each software addition and modification is carefully analyzed for its Source Line of Code (SLOC) and performance impact and benefit to the overall product. These fundamental software engineering concepts and complementing processes ensure the

CoreDX DDS baseline maintains it status as the World Leading Small Footprint DDS Implementation.

The complete CoreDX DDS baseline includes fewer than 35,000 SLOC for the Standard Edition. A Safety Critical Baseline for the CoreDX DDS product has fewer than 13,000 SLOC. This Safety Critical Baseline includes all the QoS policies and features of the standard CoreDX DDS baseline.

CoreDX DDS is Interoperable:

DDS Interoperability is the ability of DDS implementations from different vendors to communicate[7]. The Real-Time Publish Subscribe (RTPS) protocol defines the standardized wire protocol for DDS and is what allows interoperability on the wire between different implementations of DDS. CoreDX DDS can "talk" with every other type of DDS. The wire protocol is standardized; ensuring programs using different DDS products can discover each other and exchange data and can communicate.

There are a growing number of vendors active in interoperability testing and demonstration, including Twin Oaks Computing, RTI, and PrismTech. These vendors are active at the OMG Technical Meetings and regularly test and demonstrate their interoperable DDS products. [8] Twin Oaks brings DDS interoperability to Android embedded platforms.

CoreDX DDS applications can easily communicate with applications based on DDS from other vendors. This multi-vendor interoperability is enabled by multiple standards managed by the Object Management Group (OMG), including specifications of the application programming interface (API), real-time publish subscribe wire protocol (RTPS), and quality of service (QoS) features[9]. CoreDX DDS includes proven support across all of these interoperability aspects.

DDS consumers recognize the benefits of interoperability, especially where it provides them the flexibility to adapt and extend their systems with very little cost. Here are a few examples of the types of new devices are our clients are using or planning to use to extend their projects with CoreDX DDS:

- Android based phones, tablets and embedded devices
- QNX based mobile devices
- Set-top boxes
- Gateways
- Gumstix tiny Linux computers
- Micrium µC OS
- FPGA's
- Safety Critical Applications

[7] http://www.omg.org/technology/documents/dds_spec_catalog.htm
[8] http://www.omg.org/news/releases/pr2009/03-25-09.htm
[9] http://www.omg.org/technology/documents/dds_spec_catalog.htm

QoS Policies to Tailor Communications Behavior

DDS provides a rich set of Quality of Service (QoS) policies to tailor the behavior of communications. These QoS policies can be used individually or together to affect a variety of communications aspects, including reliability, performance, persistence of data, and amount of system resources used. The breadth and depth of the configuration available with these QoS policies allow CoreDX DDS to be a superior choice for communications in a large variety of industries and architectures.

Examples of QoS policies include

- **Reliability** (what are the reliability requirements for this data?)
- **Durability** (how long is data saved for possible future publication?)
- **History** and **Resource Limits** (what are the storage requirements?)
- **Filtering** and **Presentation** (which data should be presented to the subscriber, and how?)
- **Ownership** (are there any failover or redundancy requirements?)

These are just a few of the twenty-two distinct QoS policies defined by the DDS standards. However, it is the **coverage** of these QoS policies – the number of these standardized QoS policies implemented by each DDS vendor – that allows for truly interoperable implementations. All of these interoperability aspects put together allow the greatest flexibility for middleware consumers.

Dynamic Type Technology

A feature exclusive to CoreDX DDS is support for Dynamic Type Technology. This innovative new technology eases system integration challenges, and enables bridging DDS data between disparate systems in a flexible and dynamic environment. This technology enables DataReaders to dynamically, at run-time, determine the topic data types. Through Dynamic Type introspection, the subscribing application can explore the data type and access data fields.

Dynamic Types offer a flexible solution that lowers Total Cost of Ownership. DDS provides Dynamic Discovery of publishers and subscribers. Dynamic Discovery also makes your DDS applications extensible. This means the application does not have to know or configure the endpoints for communications because they are automatically discovered by DDS. This dynamic discovery goes even further than discovering endpoints. DDS will discover if the endpoint is publishing data, subscribing to data, or both. It will discover the type of data being published or subscribed to. It will also discover the publisher's offered communication characteristics and the subscriber's requested communications characteristics. All of these attributes are taken into consideration during the dynamic discovery and matching of DDS participants.

CoreDX DDS and Android Sample Case Studies:

ContextNet and Android:

The Laboratory for Advanced Collaboration (LAC) chose to utilize the Twin Oaks Computing CoreDX DDS University Licensing Program for an ongoing project: ContextNet. Project ContextNet aims at enabling communication services for large and wide scale exchanges, including on-line monitoring or coordination of mobile device activities, and information sharing through social networks. These entities may be users of portable devices (e.g. smartphones), vehicles, or moveable gadgets.

CoreDX DDS is being used to build the backbone of the communication infrastructure for the project, which will run via independent users in diverse network domains. This infrastructure will communicate with approximately 30,000-50,000+ mobile devices simultaneously, each of them sending data every 30 seconds.

ContextNet is primarily focused on addressing three major challenges:

- Enabling the scalable distribution of information among hundreds of thousands of context-producing and context-consuming entities
- Devising automated reasoning techniques that are inherently distributed and capable of detecting application-relevant patterns of global context situations (e.g. identify over- or underload conditions in the distribution density of the mobile entities).
- Using semantic Web to combine several types of context (computing, physical, time, user context) and integrate it with social networks so as to leverage the communication and coordination capabilities of mobile users and/or vehicles.

Twin Oaks Computing supplied the University with CoreDX DDS: the middleware they needed to facilitate communications. CoreDX DDS is a high performance, robust and scalable data-centric publish-subscribe peer-to-peer architecture for real time data distribution. CoreDX DDS provides a wide set of configurable Quality of Service (QoS) policies for tailoring the communication behavior between producers and consumers of data. Some of the benefits of CoreDX DDS include decoupling software components, high availability, interoperability between implementations, and automatic discovery of comparative communication peers.

Consider an example DoD system:

DCS Corp, a company that works closely with government agencies in the national security sector, recently chose Twin Oaks Computing's CoreDX DDS for their unmanned robotic systems project.

DCS Corp is developing a Graphical User Interface (GUI) that controls these unmanned robotic systems. Their software architecture uses DDS to communicate user interactions, events, and status information between graphical displays (controllers) and the unmanned systems. Their software was originally developed with DDS and developed to run on Windows and Linux platforms, but DCS Corp recently began exploring porting to Android devices as well. The challenge they faced was to port their existing C++ code base to Android and minimize

redevelopment so that they could maintain consistency across platforms. Their current DDS provider did not port to Android, but due to the standardized API and interoperability of DDS, DCS Corp was successfully able to migrate to CoreDX DDS for their project.

> *"CoreDX was the only vendor to provide a DDS distribution for Android and it allowed us to migrate to CoreDX DDS without significant changes to our DDS interface software." – Brian Wood, DCS Corp*

Twin Oaks Computing provided additional support for their effort by preparing and releasing a C and C++ binding for their CoreDX DDS Android distribution. This allowed DCS Corp to use the Android Native Development Kit (NDK) to develop C++ applications instead of requiring them to use Java, saving DCS Corp development time.

> *"The small footprint of CoreDX DDS was also significant due to the embedded nature of our software and it has performed well on our small Android devices"*
> *– Brian Wood, DCS Corp.*

The Power of Interoperability

The Android market and devices are proliferating at an amazing rate, and many existing DDS users would like to make use of this technology by extending the reach of their existing systems to individual, mobile, Android devices.

The original DDS vendor does not support Android, but because DDS is a **Standards Based Technology** with proven **Interoperability**, this system maintainer can look to other DDS vendors for a possible solution. In this particular example, the contractor maintaining this DoD system contacted Twin Oaks Computing and CoreDX DDS with the hopes of finding a native DDS solution for Android that would meet the customer's requirements - without requiring them to replace their existing DDS solution. Because of DDS Interoperability, they were successful. Now the customer has their enhanced system, connecting their legacy components with new Android devices, and they were able to do it ***without any modifications to the communication components of their legacy system.*** This is the strength of Interoperability, and the strength of the DDS Standards.

Why is CoreDX DDS the best Communications Middleware for Android?

CoreDX DDS provides a ***native DDS Android solution***: CoreDX DDS does not require gateways, translators, or web servers. Rather, native Android libraries are linked to your Android app.

CoreDX DDS is the leading small footprint implementation of the Data Distribution Standard (DDS): The full feature set of CoreDX DDS is easy to use with Size, Weight, and Power (SWaP) constrained applications such as Android. With a small footprint and full Quality of Service coverage, CoreDX DDS is designed specifically to meet the performance and complexity requirements of real-time, embedded, time-critical, and

mission-critical applications, while still being small in size and conservative in memory usage.

Core DX DDS has small run time requirements: CoreDX DDS can be used in a wide variety of embedded applications with minimal memory and CPU resources, reducing the amount of static memory (or FLASH) required to store your application. Based on an anonymous survey of Twin Oaks Computing customers conducted between December 2011 and January 2012, the features clients found most useful were the minimal run time memory footprint, ease of use, and small library size CoreDX DDS offers.

CoreDX DDS is easy to use: CoreDX DDS has a clean, easy to use Application Programming Interface (API), uncluttered by any unnecessary or confusing configuration parameters. CoreDX DDS features completely native source code with no 3rd party products or packages, and is written to the DDS standards. This translates into clean source code, with low Software Line of Code (SLoC) counts.

CoreDX DDS supports advanced reliable communications technology: CoreDX DDS can easily be employed reliably in wireless and other unreliable network environments (perfect for Android!). CoreDX DDS has lightweight, reliable communications protocols that have higher efficiency and scalability than TCP.

CoreDX DDS has proven vendor interoperability: CoreDX DDS can exchange data and communicate with every other implementation of DDS.

CoreDX DDS supports multiple development languages and environments: The same CoreDX DDS API, the same familiar programming languages, the same advanced DDS features are all available for Android. CoreDX DDS applications run on your favorite Android smart phone, tablet, and other embedded computers; we support Android on all the common (and some uncommon) hardware platforms. C, C++, and Java languages are supported for Android.

Key Points:

- Communications Middleware is computer software that enables two otherwise separate software components, processes, and/or applications to exchange information.
- The purpose of Communications Middleware is to simplify the designing, programming, and managing of software applications by streamlining the way these applications receive and process data.
- DDS is a communications middleware, in charge of transferring information
- DDS simplifies software systems, and reduces risk and cuts costs through development, integration, deployment, and lifetime maintenance of distributed software systems.
- Systems that use DDS to communicate can do so independently of each other.
- DDS automatically knows how to send and receive messages with other DDS users
- DDS participants can be on the same machine or across a network
- DDS ensures consistency
- DDS has no single point of failure
- DDS can be used wirelessly to communicate information
- DDS is reliable and always available
- CoreDX DDS is the leading small footprint implementation of the Data Distribution Standard (DDS)
- CoreDX DDS is easy to use, has small run time requirements, and a low line of code count
- Core DX DDS provides a rich set of Quality of Service (QoS) policies
- Core DX DDS has small run time requirements
- CoreDX DDS is easy to use
- CoreDX DDS has proven vendor interoperability
- CoreDX DDS supports multiple development languages and environments

Conclusion and Summary

Communications Middleware is computer software that enables two otherwise separate software components, processes, and/or applications to exchange information, either within one device, or between multiple devices.

Data Distribution Service (DDS) is a type of Communications Middleware that simplifies software systems, and reduces risk and costs through development, integration, deployment, and lifetime maintenance of distributed software systems. DDS is now available for Android devices.

DDS technology increases software development productivity, reduces risk, and eases deployment and maintenance challenges in dynamic systems. DDS Interoperability allows consumers to replace or augment one DDS implementation with another better suited to their requirements and extend already deployed systems with new applications using different DDS implementations. This flexibility further reduces risk and further enables management of changing systems.

With the increasing popularity of Android apps, many developers would like to make their existing applications Android compatible. CoreDX DDS is a full-featured DDS Communications Middleware implementation that comes in a surprisingly small package, perfect for Android devices. It is easy to use, has small runtime requirements, is interoperable, and supports multiple development languages and environments. The CoreDX DDS source code is clean, easy to read, easy to build, easy to port, and easy to modify.

Download a free evaluation copy at www.twinoakscomputing.com/coredx/download.

Twin Oaks Computing

Twin Oaks Computing, Inc. is a company dedicated to developing and delivering quality software solutions. Our staff has extensive experience developing and supporting robust communication architectures. We leverage this world-class technical experience to provide innovative and useful communication software systems. We build the software that collects, manages, and distributes information in a wide range of industries. Our software is in use around the world supporting critical missions.
Equally important, our clients are amazed and totally satisfied with our super responsive customer service. One of our early customers in China states,

> *"Twin Oaks Computing [provided] great porting work during very short period of time (each porting for about 2-3 weeks). This made me really appreciate the portability framework of CoreDX DDS."*
> - Mr. Huang

More recently, we received this comment from a customer in the United States,

> *"There is nothing I don't like about working with Twin Oaks Computing. In particular, working with Nina is a singular pleasure in today's world of technical support - she is very responsive and helpful."*
> - Dr. Michael Mezzino

About Twin Oaks Computing

With corporate headquarters located in Castle Rock, Colorado, USA, Twin Oaks Computing is a company dedicated to developing and delivering quality software solutions. We leverage our technical experience and abilities to provide innovative and useful services in the domain of data communications. Founded in 2005, Twin Oaks Computing, Inc delivered the first version of CoreDX DDS in 2008. The next two years saw deliveries to over 100 customers around the world. We continue

Contact

Twin Oaks Computing, Inc.

(720) 733-7906

(855)-671-8754 (toll free)

+33 (0)9 62 23 72 20

755 Maleta Lane

Suite 203

Castle Rock, CO. 80108

www.twinoakscomputing.com