CoreDX DDS Product Brief

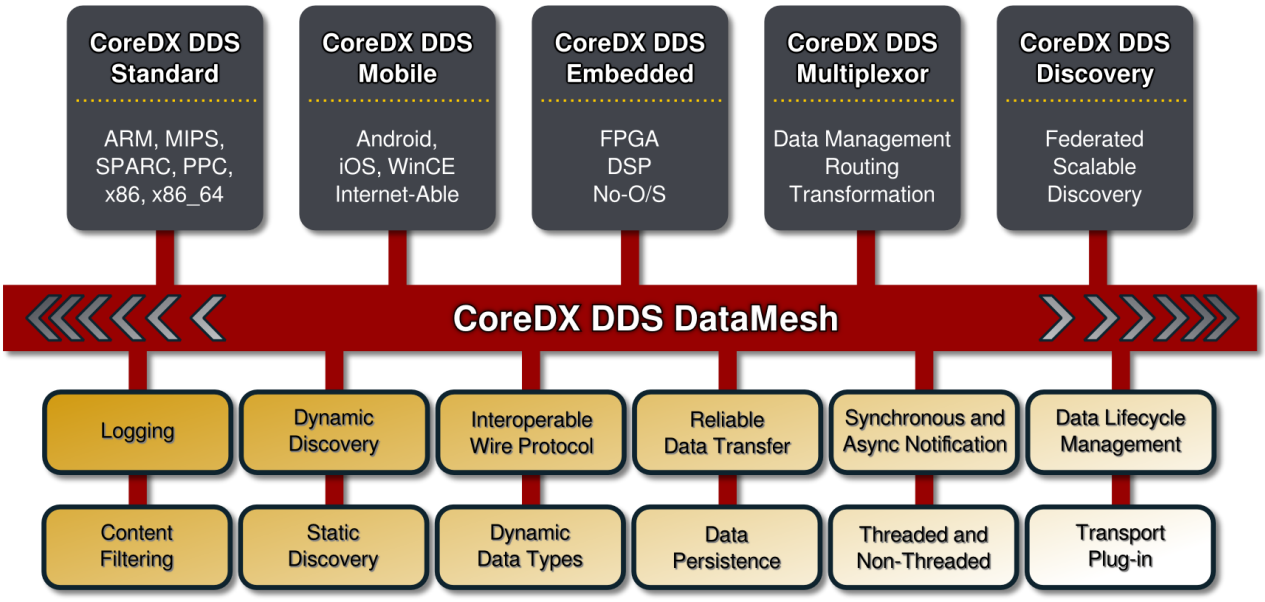
1 Overview

CoreDX DDS is a software technology that provides sophisticated high-performance, interprocess, data communication. The communication protocol is standardized to provide interoperability with external products. The technology is configurable to provide robust Quality of Service options. CoreDX DDS supports a wide range of hardware and software platforms for flexibility. CoreDX DDS is dynamic and supports a loosely coupled system architecture by discovering and responding to events on the network.

CoreDX DDS is available to support many different application domains, from large scale enterprise systems, mobile and hand-held devices, to small highly-embedded devices such as Field Programmable Arrays (FPGAs) and Digital Signal Processors (DSPs) with limited resources.

With over one hundred thousand deployments, CoreDX DDS is a mature and stable technology. CoreDX DDS operates today in defense systems, medical devices, consumer electronics, and even robots on-board the International Space Station.

The CoreDX DDS line of products includes technology to address diverse networking environments, limited capability networks, data bridging and routing, and massive scalability.



2 CoreDX DDS Features

CoreDX DDS is lightweight, but not light on features.

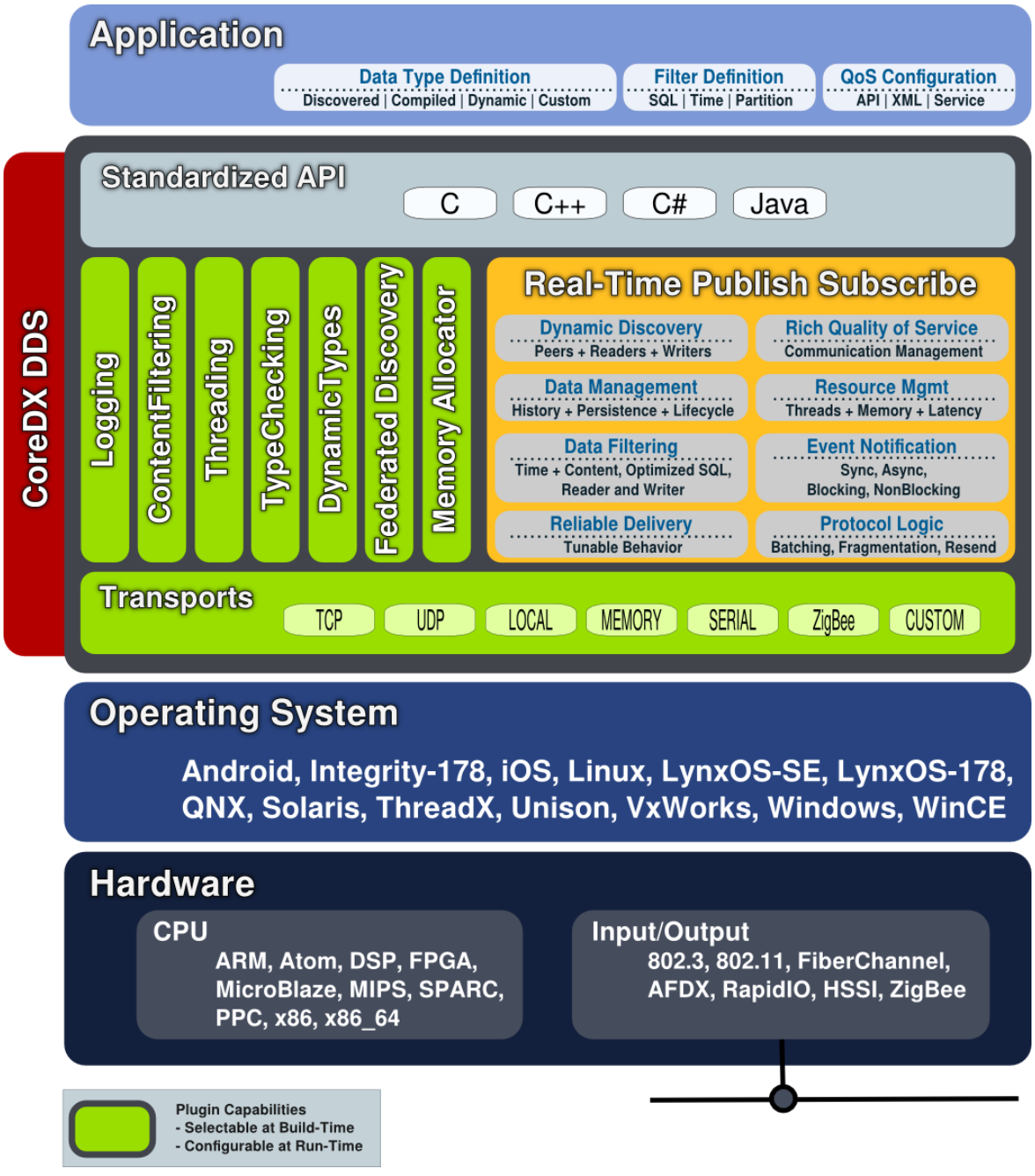
2.1 Resource Management

CoreDX DDS is developed with an emphasis on Resource Management.

The source code baseline for the core features of CoreDX DDS includes only 35-thousand Lines of Code (LOC). The resulting binary library is about 500KBytes. This makes it easy to deploy CoreDX DDS to small and resource constrained devices with limited static storage.

The run-time memory requirements of CoreDX DDS are extremely modest. A typical application requires less than 100KBytes of RAM at run-time to host CoreDX DDS.

The CPU processing requirements of CoreDX DDS are very low. CoreDX DDS is built to avoid busy-waiting, and therefore does not waste power or CPU cycles. When not actively sending or receiving data, the CoreDX DDS middleware will not burden the CPU, and CPU utilization is essentially zero.



2.2 Performance

Throughput: 1GBit Ethernet: up to 1000 Mbps

Latency: on platform: less than 22 µsec latency; 1Gbit Ethernet: 45 µsec

Further, the standard deviation in the latency measurements is exceptionally small: less than 2 µsecs. This shows that CoreDX DDS can deliver low latency data with very little 'jitter' or variation.

2.3 Application Programming Interface (API)

The standardized interface of CoreDX DDS supports multiple programming languages, including C, C++, C#, and Java. The API is well documented and extremely easy to use. Several application programs are provided with the CoreDX DDS product to illustrate the API usage.

2.4 Data Types

CoreDX DDS supports the processing of language-natural data types (for example: structures) and provides robust type-checking to avoid errors during development and deployment. Application defined data-types can be specified in many different ways.

1. Design-Time data type construction: Data types can be specified at design time, using the Data Definition Language (DDL) – a language-neutral syntax. These DDL based data types are 'compiled' to generate language specific support files for integration into the application.
2. Run-Time data type construction: Data types can be built dynamically in the application source code. This is useful if the application data types are influenced by outside circumstances at run-time.
3. Discovered data types: During run-time, a CoreDX DDS application can discover data types from peers. A Discovered data type can be used to initialize a Dynamic data type.

In any case, whether defined by DDL, built dynamically at run-time, or discovered at run-time, all of these data types can be mixed and used interchangeably.

2.5 Strong Type Checking

CoreDX DDS supports type safety during code development. The application specifies the Types of data that will be used for communications, and these types are passed into write() calls and returned from read() calls. Because the CoreDX DDS middleware supports programming with data types that are natural for the application, many common programming errors can be automatically identified by the compiler. Strong Type Safety reduces programming errors, since type mismatches are discovered at compile time.

CoreDX DDS can perform type checking at run-time to detect and identify type mismatches among deployed entities. This type-checking maintains the integrity of applications at run-time. The type checking is automatic and the application an be notified of any data type incompatibilities.

2.6 Plug-in Feature Flexibility

2.6.1 Logging

CoreDX DDS supports configurable logging to trace events within the middleware. Logging can be selected at build time and configured at run-time. The set of logged events can be selected, and include Errors, Discovery, Data, Factory, Liveliness, Status, Transport, and Trace.

2.6.2 Content Filtering

Data conveyed by the CoreDX DDS middleware can be filtered based on data content and data timing. The capability to filter data based on content is selectable at build time and configurable at run-time. Content filtering is specified by an SQL query that identifies data fields and ranges. The SQL operators supported by CoreDX DDS include NOT, AND, OR, LESS_THAN, GREATER_THAN, EQUAL, LIKE, and IN. The 'IN' operator is an optimized query to support selection of data from a range of options – it supports a very high-performance Complex Event Processing (CEP) application.

2.6.3 Threading

There are multiple threading options available within CoreDX DDS – from non-threaded to multiple thread-domains. The non-threaded approach provides the best performance on low-powered devices by avoiding all locking primitives and eliminating all application level context switches. This improves cache dynamics and supports reduced system latency.

2.6.4 Federated Discovery

An important feature of CoreDX DDS is the ability to dynamically establish connections and replicate data between peers. By default, the discovery process is a peer to peer architecture, providing a robust and resilient mesh of data communication. However, in some instances, the overhead associated with this approach is prohibitive or unacceptable: the growth is exponential with respect to the number of peers.

To address these scenarios, CoreDX DDS offers a Federated Discovery component, in which the discovery interconnection is not simply peer-to-peer. The Federated Discovery agent performs peer matching and establishes communication on an as-needed basis, and can scale to very large systems while dramatically reducing the network and memory resources required.

2.6.5 Memory Allocation

CoreDX DDS supports alternative memory allocation schemes. By default, memory required by the middleware is allocated from the Operating System. In those case where this facility is not available or desirable, CoreDX DDS can be configured to operate within a fixed memory region.

2.7 Real Time Publish Subscribe

At the heart of CoreDX DDS is an implementation of the interoperable Real-Time Publish Subscribe (RTPS) protocol. This protocol provides advanced Inter-Process Communications with support for data

caching, history, persistence, guaranteed delivery, lifecycle management, filtering, and many other features. The RTPS implementation within CoreDX DDS has field-proven interoperability along with advanced features not found in other implementations.

The capabilities, performance, and behavior of the RTPS core is configurable through a rich set of Quality of Service policies.

2.8 Transports

The architecture of CoreDX DDS supports a wide variety of underlying transport mechanisms. The availability of transports can be configured at build-time, with further configuration performed at run-time. Furthermore, custom transports are supported and can be integrated at run-time through a simple API.

CoreDX DDS supports multiple simultaneous transports, and can select among the available configured transports to achieve communication.

2.9 Operating Systems and Hardware Platforms

The wide range of Operating Systems and Hardware platforms supported by CoreDX DDS is evidence of the extreme portability of the software. This portability is the result of concentrated engineering efforts to issolate and eliminate dependencies on the operating system and hardware. In fact, CoreDX DDS has been deployed to platforms that provided little more than a clock, memory, and interrupts.

3 Customization

The lightweight, small footprint of CoreDX DDS makes it very easy to customize. The engineers at Twin Oaks Computing can provide engineering services to address unique project requirements, from custom data transports to custom operating system ports. Furthermore, the low line-of-code count and clean architecture means that project engineers can perform customizations as well.

For example, the API to integrate a custom transport includes fewer than twenty functions that must be implemented. It is clear how an engineer could develop a custom transport by modeleing on one of the existing transports in the current code baseline.

4 Conclusion

CoreDX DDS is a powerful technology that facilitates the development of flexible and dynamic Open Architecture applications. Built on standardized protocols and interfaces, CoreDX DDS enhances productivity and reduces maintenance costs. Give it a try today – evaluations are free for download on our web site!