PRISMTECH
Productivity Tools & Middleware

OpenSplice Overview

White Paper

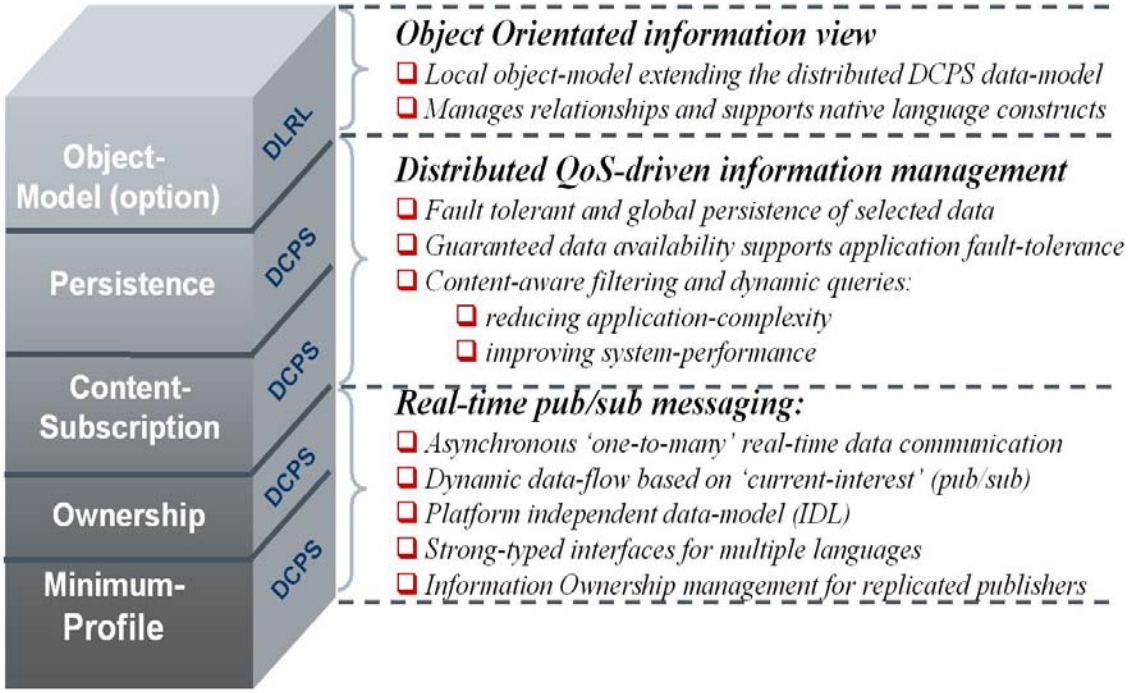
Hans van't Hag, Product Manager

1 Overview

In the large class of network-centric systems, real-time availability of information is of utmost importance. Information generated from multiple sources must be distributed and made available to 'interested parties' taking into account Quality of Service (QoS) offerings by information-producers and requests by information-consumers. Especially in real-time and mission-critical systems, getting *'the right data at the right time at the right place'* is not a trivial task at all and up until recently, there were no standards nor COTS products that addressed this challenge in an integrated solution. The **OMG** recognized this need for a Data Distribution Service (DDS) and organized members with vast experience in both the 'underlying' technologies (networking and information-management) as well as 'user-level' requirements (distributed, real-time and mission-critical system characteristics), including Thales Naval Netherlands, to join forces and these members defined the **'OMG-DDS'** service. The OMG-DDS service specifies a coherent set of profiles that target real-time information-availability for domains ranging from small-scale embedded control systems up to large-scale enterprise information management systems. Each DDS-profile adds distinct capabilities that define the service-levels offered by DDS in order to realize this ***'right data at the right time at the right place'*** paradigm:

- **Minimum Profile**: this '*basic*' profile utilizes the well-known publish/subscribe paradigm to implement highly efficient information-dissemination between multiple publishers and subscribers that share interest in so-called '**topics**'. Topics are the basic data-structures expressed in the OMG's IDL-language (allowing for automatic generation of typed 'Readers' and 'Writers' of those 'topics' for any mix of languages desired). This profile also includes the QoS framework that allows the middleware to 'match' requested and offered Quality of Service parameters (the minimum profile offering basic QoS attributes such as 'reliability', 'ordering' or 'urgency').
- **Ownership Profile**: this '*replication*' profile offers support for replicated publishers of the same information by allowing a '**strength**' to be expressed by each publisher so that only the 'highest-strength' information will be made available to interested parties.
- **Content Subscription Profile**: this '*content awareness*' profile offers powerful features to express fine-grained interest in specific information content (content-filters). This profile also allows applications to specify **projection-views** and **aggregation** of data as well as dynamic **queries** for subscribed 'topics' by utilizing a subset of the well-known SQL language whilst preserving the real-time requirements for the information-access.

- **Persistence Profile**: this '*durability*' profile offers transparent and fault-tolerant availability of 'non-volatile' data that may either represent persistent 'settings' (to be stored on mass-media throughout the distributed system) or 'state' preserved in a fault-tolerant manner outside the scope of transient publishers (allowing late-joining of applications and dynamic re-allocation).
- **DLRL Profile**: this '*object model*' extends the previous four data-centric 'DCPS' profiles with an **object-oriented view** on a set of related topics thus providing typical OO-features such as navigation, inheritance and use of value-types.



OMG-DDS LAYERS

2 OpenSplice Summary

PrismTech's OpenSplice, is a 2nd generation fully compliant OMG-DDS implementation, offering support for all the DCPS profiles (minimum profile, ownership profile, content subscription profile and persistence profile) as well as the DLRL object-profile. OpenSplice was initially developed as SPLICE-DDS by Thales Naval Netherlands (TNL), one of the co-authors of the DDS specification and is the result of TNL's over 15-year experience in developing distributed information systems for naval Combat Management Systems (CMS). This field proven middleware is used as the 'information backbone' of TNL's TACTICOS CMS currently deployed in 15 navies around the world. OpenSplice is the 2nd generation COTS evolution of this successful product and consists of several modules that cover the full OMG specification as well as provision of total-lifecycle support by an integrated Productivity toolsuite:

- **OpenSplice core modules** cover the "Minimum" and "Ownership" profiles that provide the basic pub-sub messaging functions. The minimum profile is meant to address real time messaging requirements, where performance and low footprint are essential. The ownership profile provides basic support for replicated publishers where 'ownership' of published data is governed by 'strength' indicating the quality of published information.
- **OpenSplice "content subscription" and "persistence" profiles** provide the additional information management features, key for assuring high information-availability (fault-tolerant persistence of non-volatile information) as well as powerful 'content-aware' features (filters and queries), thus enabling unmatched performance for the full range of small-scale embedded up to large scale fault tolerant systems.
- **OpenSplice "Data Local Reconstruction Layer" profile** provides an object-oriented view on a set of related topics allowing for typical OO-features such as value-types with inheritance, object-navigation and others. This 'DLRL' profile of the DDS-spec as well as its OpenSplice implementation is still a 'work-in-progress' and first implementations are expected in Q4 2006.
- **OpenSplice "Productivity tool suite"**, a complete Eclipse-based productivity toolsuite modeled after PrismTech's successful MDA Spectra product-line supporting information-modeling, object-modeling, application-modeling, DLRL and DCPS code generation and runtime system-monitoring from within an integrated environment (available Q4'2006)

Free evaluation licenses of OpenSplice are available by e-mailing sales@prismtech.com. Currently supported platforms include Solaris / Sparc, Linux / x86, Windows / X86 and VxWorks / PowerPC, whereas supported languages are C, C++ (in seamless co-habitation with Any ORB and related C++ compiler) and Java.

3 OpenSplice Architecture

Overall:

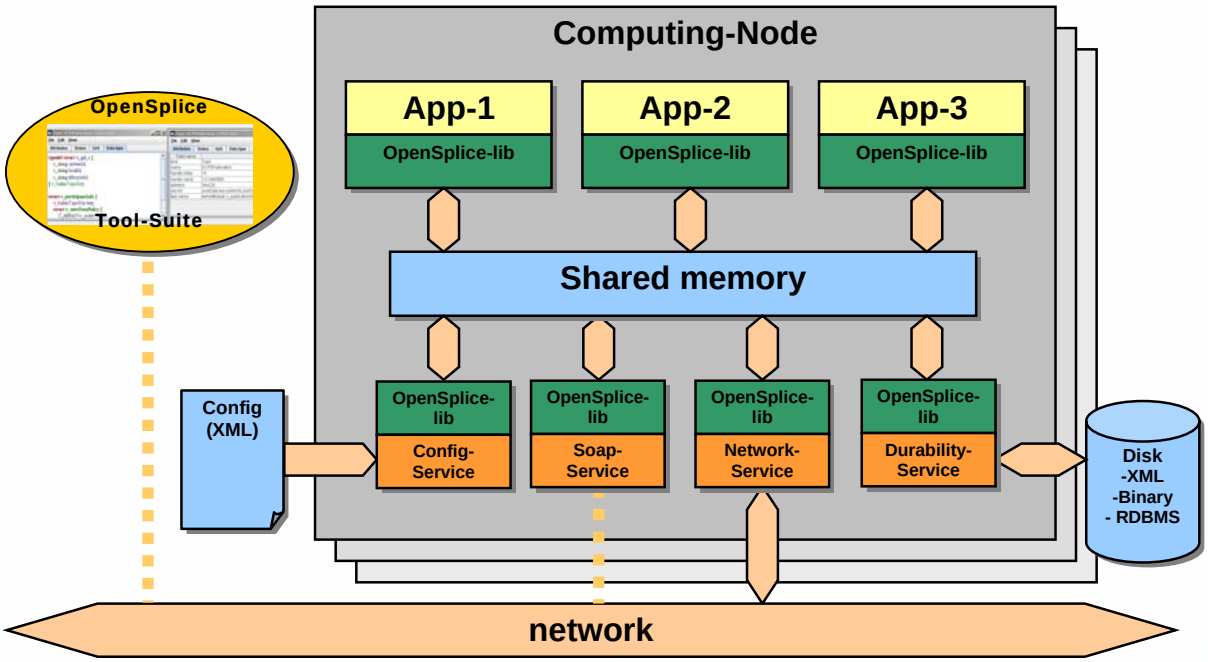
To ensure scalability, flexibility and extensibility, OpenSplice has an internal architecture that utilizes shared-memory to 'interconnect' not only all applications that reside within one computing node, but also 'hosts' a configurable and extensible set of services. These services provide 'pluggable' functionality such as networking (providing QoS-driven real-time networking based on multiple reliable multicast 'channels'), durability (providing fault-tolerant storage for both real-time 'state' data as well as persistent 'settings'), and remote control & monitoring 'soap-service' (providing remote web-based access using the SOAP protocol from the OpenSplice Tuner tools).

Scalability:

OpenSplice utilizes a shared-memory architecture where data is physically present only once on any machine, and where smart administration still provides each subscriber with his own private 'view' on this data. This allows a subscriber's data cache to be perceived as an individual 'database' that can content-filtered, queried, etc. (using the content-subscription profile as supported by OpenSplice). This shared-memory architecture results in an extremely low footprint, excellent scalability and optimal performance when compared to implementations where each reader/writer are 'communication-endpoints' each with its own storage (i.e. historical data both at reader and writer) and where the data itself still has to be moved, even within the same platform.

Configuration:

The OpenSplice middleware can be easily configured 'on the fly' by specifying (only the needed) services to be used as well as configuring those service for optimal matching with the application domain (networking parameters, durability levels, etc). Easily maintainable XML-file(s) are utilized to configure all OpenSplice services. OpenSplice configuration is also supported by means of the MDA toolset allowing system/network modeling and automatic generation of the appropriate XML configuration files.



OpenSplice Pluggable Service Architecture

(only 1 node is shown, typically there are many nodes within a system)

4 OpenSplice DDS implementation Benefits

The table below shows the following aspects of OpenSplice:

- **Features**: significant characteristics of the product
- **Advantages**: explaining why a feature is important
- **Benefits**: how users of OpenSplice can exploit these advantages

	FEATURE	ADVANTAGE	BENEFIT
GENERAL	**Information-centric**	Enable dynamic, loosely coupled syst.	*Simplified & better scalable architectures*
	Open standard	'Off-the-shelf' solutions	*Lower cost, no vendor-lock-in*
	Built on proven tech.	Meant for most demanding envnts.	*Assured quality and applicability*
	TNN/PT 'heritance'	Decade long of 'DDS experience'	*Proven suitability in mission-critical domain*
FUNCTIONAL	**Real-time pub/sub**	Dynamic/asynchroneous data comm.	*Autonomous de-coupled applications*
	Persistence profile	Fault tolerant data-persistence	*App. fault-tolerance and data high-availability*
	Content-sub. Profile	Reduced complexity & higher perf.	*Easier application design & scalable systems*
PERFORMANCE	**Shared-memory**	low footprint, instant data-availability	*Processor Scalability*
	Smart networking	Efficient data-transport	*Network Scalability*
	Extensive IDL sup.	Includes unbounded strings,sequences	*Data Scalability*
USABILITY	**Multiple language**	Any (mix) of C, C++, Java, Ada	*Supports (legacy) code, allows hybrid systems*
	Multiple platforms	Any (mix) of Enterprise & RTE Oss	*Intercons enterprise and embedded systems*
TOOLING & EASE-OF-USE	All meta-data at runtime	Dynamic discovery of all 'entity-info'	*Guaranteed data-integrity*
	Powerful tooling	Support for complete system lifecycle	*Enhanced productivity and System Integration*
	Remote connect	Web-based remote access & control	*Remote diagnostics using standard protocols*
Legend:	EQUAL to competition	BETTER than competition	FAR-SURPASSING competition

OpenSplice F.A.B Table

5 OpenSplice Productivity Tools

PrismTech has established an excellent reputation for creative and innovative Model Driven Development (MDD) tools integrated with runtime tools that support the design, development and maintenance of systems and applications built using our products.

OpenSplice provides an integrated toolchain consisting of the design-time information modeling tool called the “OpenSplice Information Modeler”, an application-modeling and code-generation tool called “OpenSplice Application modeler”, and a runtime monitoring and management tool called the “OpenSplice Tuner”. These tools are all based on the domain-specific model driven techniques used within PrismTech’s other successful product ranges, such as the Spectra Tools targeting the SDR domain.

The clear aim of these tools provided with OpenSplice is to facilitate proper information modeling, reduce application complexity, shorten application development time, improve application quality and finally ensure standards compliance. Support is provided for developers who are developing in a top-down manner, starting from a UML model and using DLRL, as well as users who are developing bottom-up with an existing DCPS (Topic) model that potentially wish to migrate to a higher level DLRL model. System-integration and evolution are supported by the runtime Tuner tool offering remote-access capabilities and fully integrated with the design-time modeling tools thus supporting real ‘round-trip’ engineering facilities.

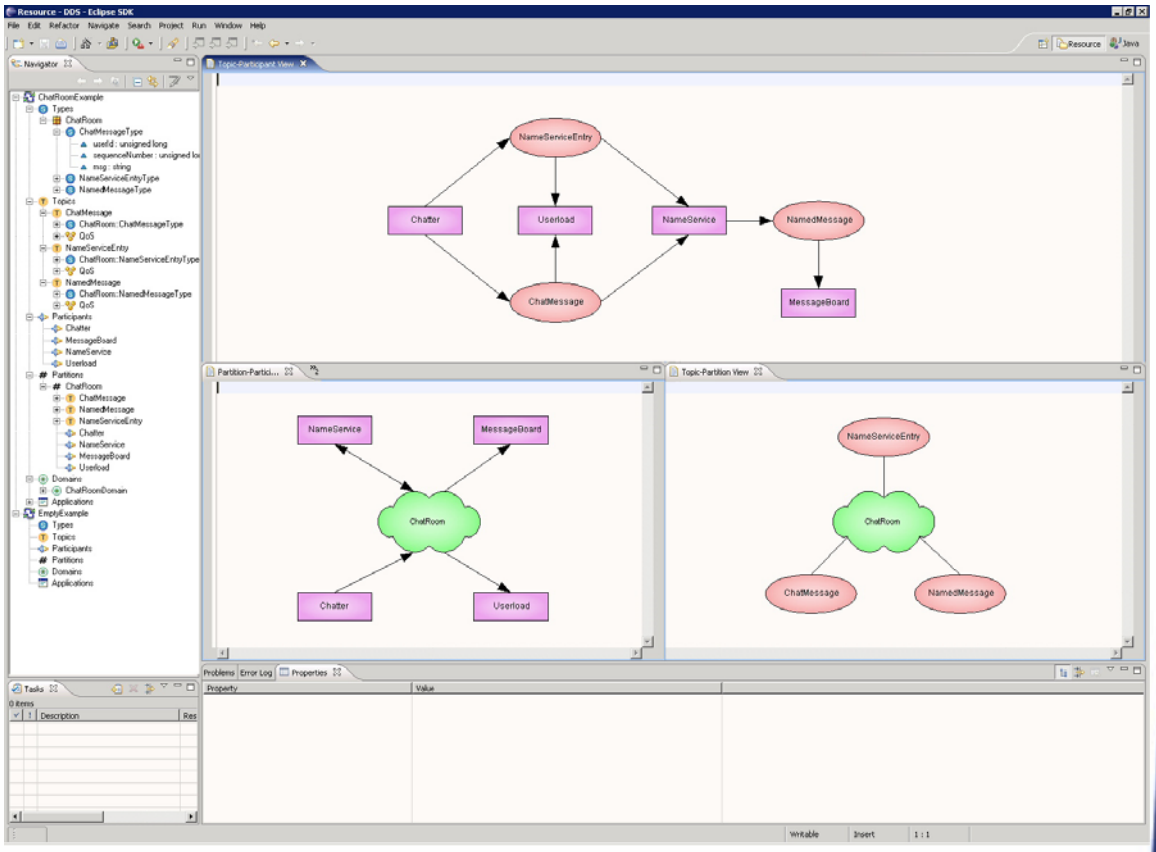
OpenSplice Information and Application Modeler tools

The purpose of these tools is to facilitate DDS-based system development by clearly distinguishing between the various ‘scopes’ and ‘lifecycle stages’ of the system supported by visual composition, configuration and round-trip engineering. This purpose is realized by the following means:

- **Guidance**: the tools provide context-aware guidance regarding the overall DDS concept, patterns and best-practices.
- **Well-defined hierarchical steps**: such as information modeling (topic definitions in IDL, code-generation for topic QoS), application design (code-generation for application frameworks and DDS entities such as publishers/writers, subscribers/readers) and system deployment (information partitioning, network-configuration and durability configuration resulting in XML-based OpenSplice configuration data).

These tools will become available during Q3 and Q4 of this year, offering the following benefits:

- **Simplified integration** with other tools: the tools will be delivered as plug-ins for the popular Eclipse Tool Framework thus enabling straightforward integration with other eclipse-based tools, including PrismTech's Spectra Tools.
- **Intuitive visual development** of DDS-based application through a comprehensive graphical user interface.
- **Remove defects and errors** that would otherwise be found only at runtime through the exploitation of a DDS Meta Model, a comprehensive and complete model that includes DDS domain specific constraints.
- **Provide domain-specific as well as UML-based views** for system-level information modeling, DLRL object modeling, Application modeling, as well as runtime deployment (covered by the OpenSplice tuner described below).



OpenSplice Information & Application Modeler Preview

OpenSplice Tuner

The 100% Java based OpenSplice Tuner tool greatly aids the design, implementation, test and maintenance of OpenSplice based distributed systems (the OpenSplice Tuner is both available as a 'stand-alone' java-program as well as an Eclipse plug-in for the Productivity tool suite):

- **Design**: During the design phase, once the information model is established (i.e. topics are defined and 'registered' in a runtime environment, which can be both a host-environment as well as a target-environment), the OpenSplice Tuner allows creation of publishers/writers and subscribers/readers on the fly to experiment and validate how this data should be treated by the middleware regarding persistence, durability, latency, etc.

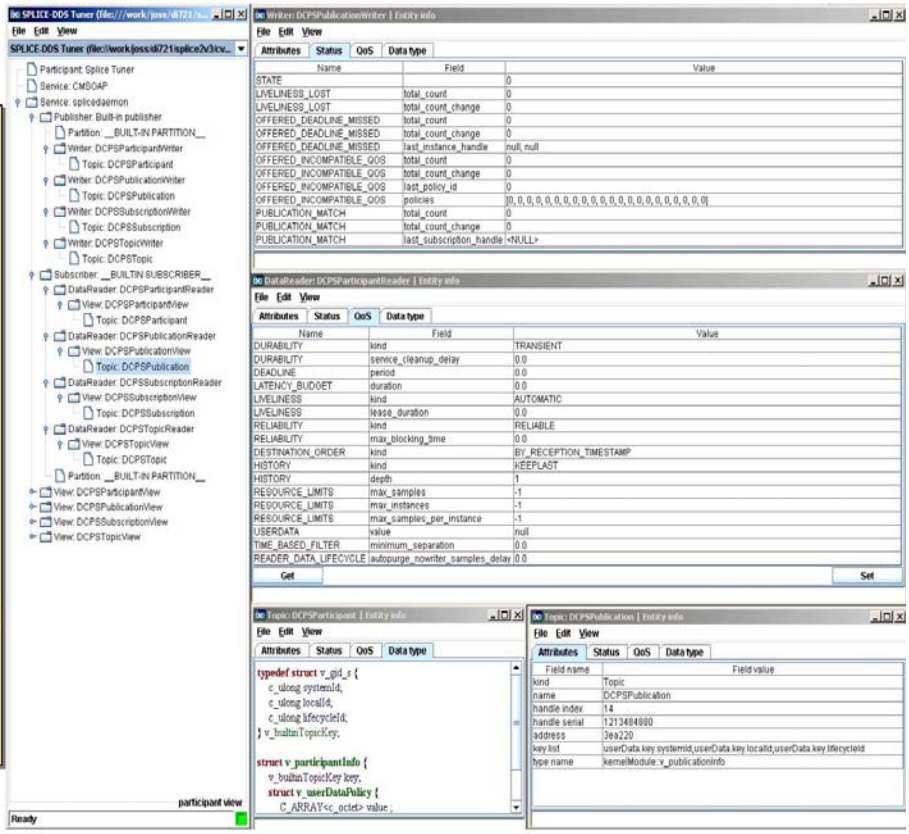
- **Implementation**: During the implementation phase, where actual application-level processing and distribution of this information is developed, the OpenSplice Tuner allows injection of test input-data by creating publishers and writers 'on the fly' as well as validating the responses by creating subscribers and readers for any produced topics.

- **Test**: during the test phase, the total system can be monitored by inspection of data (by making 'snapshots' of writer- and reader-history caches) and behavior of readers & writers (statistics, like how long data has resided in the reader's cache before it was read) as well as monitoring of the data-distribution behavior (memory-usage, transport-latencies).

- **Maintenance**: Maximum flexibility for planned and 'ad-hoc' maintenance is offered by allowing the 100% JAVA based OpenSplice Tuner tool-suite (which can be executed on any JAVA enabled platform without the need of OpenSplice to be installed) to remotely connect via the web-based SOAP protocol to any 'reachable' OpenSplice system around the world (as long a HTTP-connection can be established with the OpenSplice computing-nodes of that system). Using such a dynamic-connection, critical data may be logged and data-sets may be 'injected' into the system to be maintained (such as new settings which can be automatically 'persisted' using the QoS features as offered by the 'persistence-profile supported by OpenSplice).

Splice-Tuner
TOTAL SYSTEM CONTROL

- 100 % **Java-based**
- **Remote connect** via SOAP
- **Monitor & Control**:
  - all DDS-entities & relations
  - all QoS settings
  - all services such as:
    - communication
    - durability-service
- **Interactive browsing**:
  - inspect any data-cache
  - make cache-snapshots
  - view statistics
- **Reading/Writing data**:
  - create readers/writers
  - read/write any data
- **Multiple views**:
  - participant view
  - topic view
  - partition view
- **Dynamic creation** of:
  - readers (with filters/queries)
  - writers (with input validation)
- **Automatic discovery** of:
  - Partitions & participants
  - Topics with name/type
  - related publishers/writers
  - related subscribers/readers



OpenSplice Tuner

6 Conclusion

PrismTech's OpenSplice DDS product complemented by the OpenSplice productivity tool suite together encompass the industry's most profound expertise on the OMG's DDS and MDA standards and products.

Our DDS offering exploits the fielded heritage of TNL's SPLICE-DDS product combined with the innovative and successful modeling approach currently leveraged by PrismTech's Spectra SDR development support suite.

The result is unrivaled functional DDS-coverage and performance in large-scale mission-systems, fault-tolerance in information availability, and total-lifecycle support including round-trip engineering. A complete DDS solution to ensure a customer's successful adoption of this exciting new technology and to support delivery of the highest-quality applications with shortest time to market in the demanding real-time world.

Contacts

PrismTech can be contacted at the following address, phone number, fax and e-mail contact points for information and technical support.

Corporate Headquarters
PrismTech Corporation
6 Lincoln Knoll Lane
Suite 100
Burlington, MA
01803
USA

Tel: +1 781 270 1177
Fax: +1 781 238 1700

European Head Office
PrismTech Limited
PrismTech House
5th Avenue Business Park
Team Valley
Gateshead, NE11 0NG
UK

Tel: +44 (0)191 497 9900
Fax: +44 (0)191 497 9901

PrismTech Deutschland
PrismTech GmbH
Schönhauser Allee 6-7

10119 Berlin
Germany

Tel: +49 (0) 30 4403060
Fax: +49 (0) 30 44030678

PrismTech France
PrismTech SARL
Parc de la Fontaine de Jouvence, 4, Rue Angiboust
91460 Marcoussis
France

Tel: +33 (1) 69 015354
Fax: +33 (1) 69 015355

Web: *http://www.prismtech.com*
E-mail: *info@prismtech.com*

Notices