

White Paper

Rogue Wave SourcePro C++

Facilitating Mission-Critical Software Development



www.roguewave.com

Rogue Wave SourcePro C++: Facilitating Mission-Critical Software Development

Abstract 1

The SourcePro C++ Advantage 2

SourcePro C++ Products and Modules 3

SourcePro Core 4

Standard C++ Library Module 4

Standard C++ Library Module Features and Benefits 4

Essential Tools Module 5

Essential Tools Module Features and Benefits 6

Advanced Tools Module 6

Advanced Tools Module Features and Benefits 7

XML Streams Module 7

XML Streams Module Features and Benefits 7

Threads Module 8

Threads Module Features and Benefits 8

SourcePro DB 9

Interface Module 10

Interface Module Features and Benefits 10

Access Modules 11

SourcePro Net 11

Essential Networking Module 12

Essential Networking Module Features and Benefits 12

Internet Protocols Module 12

Internet Protocols Module Features and Benefits 13

Secure Communication Module 13

Secure Communication Module Features and Benefits 14

Web Services Module 14

Web Services Module Features and Benefits 15

SourcePro Analysis 15

Essential Math Module 16

Essential Math Module Features and Benefits 16

Linear Algebra Module 16

Linear Algebra Module Features and Benefits 17

Business Analysis Module 17

<i>Business Analysis Module Features and Benefits</i>	17
Currency Module	18
<i>Currency Module Features and Benefits</i>	18
SourcePro C++ Products Working Together: A Code Example	19
Related Publications	21
Solid Solutions from Rogue Wave Software	22

Abstract

Rogue Wave® SourcePro™ C++ offers developers robust, object-oriented C++ components designed to help deliver applications faster. Using SourcePro C++ gives development teams an advantage in creating stable and reliable applications that can grow and evolve with changing business needs.

The intuitive and consistent higher-level APIs provided by Rogue Wave SourcePro C++ shorten the developer learning curve and make it easier to maintain applications over time. Rogue Wave handles the granular details of development technologies and tedious, time-consuming maintenance tasks. Additionally, all Rogue Wave products are portable across leading platforms and databases, providing the flexibility to move applications as requirements change.

Crafted and documented by C++ experts and field-proven in tens of thousands of applications, Rogue Wave SourcePro C++ components can be relied on for mission-critical applications.

Rogue Wave SourcePro C++ includes four products:

SourcePro Core contains an extensive set of fundamental C++ components that handle many of the intricacies of the C++ language, and includes powerful and complete solutions for multithreading applications and for streaming XML and data non-intrusively. SourcePro Core provides essential functionality that drives each of the other SourcePro C++ products. For this reason, it is included with the purchase of any SourcePro C++ product. Alternatively, SourcePro Core can be purchased and used on its own.

SourcePro DB has a layered architecture that abstracts away the complexity of writing database applications, yet allows developers to drill down to the native database client libraries if needed. SourcePro DB contains database and data manipulation classes that encapsulate SQL 92 in an intuitive C++ interface that is easy to use and consistent across all supported databases and platforms.

SourcePro Net lets developers create secure or non-secure networked and Internet-enabled applications, handling the granular details of socket programming and Internet protocols to help developers deliver quality applications on schedule. SourcePro Net lets developers use their existing C++ knowledge to implement standard SOAP concepts, in order to create Web Services and extend their existing investments in C++ through XML.

SourcePro Analysis contains a full range of C++ components for solving mathematical problems in business and research, encapsulating algorithms that can be relied upon for accurate, precise calculations.

All Rogue Wave SourcePro C++ products simplify development projects and reduce development time by providing intuitive, easy-to-use, C++ interfaces to complex constructs. By using tested, reliable C++ components from Rogue Wave, developers can save both time and effort in order to consistently deliver projects on schedule.

The SourcePro C++ Advantage

Rogue Wave SourcePro C++ provides significant advantages to development teams building mission-critical C++ applications, including:

Broad functionality. Rogue Wave SourcePro C++ contains the components developers need for virtually any programming task in C++, with specialized functionality for multithreading, networking, Internetworking, security, database access and business analysis.

Simplified interfaces to complex underlying technologies. The consistent, object-oriented interfaces in SourcePro C++ shorten the learning curve, increasing developer productivity and the ability to deliver projects on schedule. Development teams can get a head start on building applications that solve business problems.

Integration and multiple platform offerings. All SourcePro C++ components work together across products, and also work with a variety of industry-leading operating systems, compilers, databases and threading libraries.

Flexible, cross-platform extensibility. SourcePro C++ interfaces are consistent across platforms, allowing developers in many cases to simply recompile and run their application on a different platform without rewriting code. This flexibility is an essential advantage in the face of changing business needs.

Extended life for C++ applications. SourcePro C++ helps development teams deliver applications that use XML and SOAP-based Web Services. With these components, companies can extend the life of their C++ applications by allowing them to communicate with systems written in different languages or running on dissimilar platforms.

Full source code available. Full source code for SourcePro C++ is available, facilitating debugging and integration with other third-party libraries.

Fully-tested and field-proven code offering the performance of C++. Rogue Wave SourcePro C++ components deliver the advantages of the C++ language in pre-built classes, allowing development teams to meet their development schedules and the stringent performance requirements of mission-critical applications.

Support and services from an established vendor. World-class technical support, mentoring, training and professional services from Rogue Wave Software support development efforts and help ensure project success.

SourcePro C++ Products and Modules

Figure 1 shows the four Rogue Wave SourcePro C++ products and their modules; all SourcePro C++ products – SourcePro DB, SourcePro Net and SourcePro Analysis – rely on the functionality in SourcePro Core and therefore work well together. Their common intuitive interface makes them easy to learn and use.

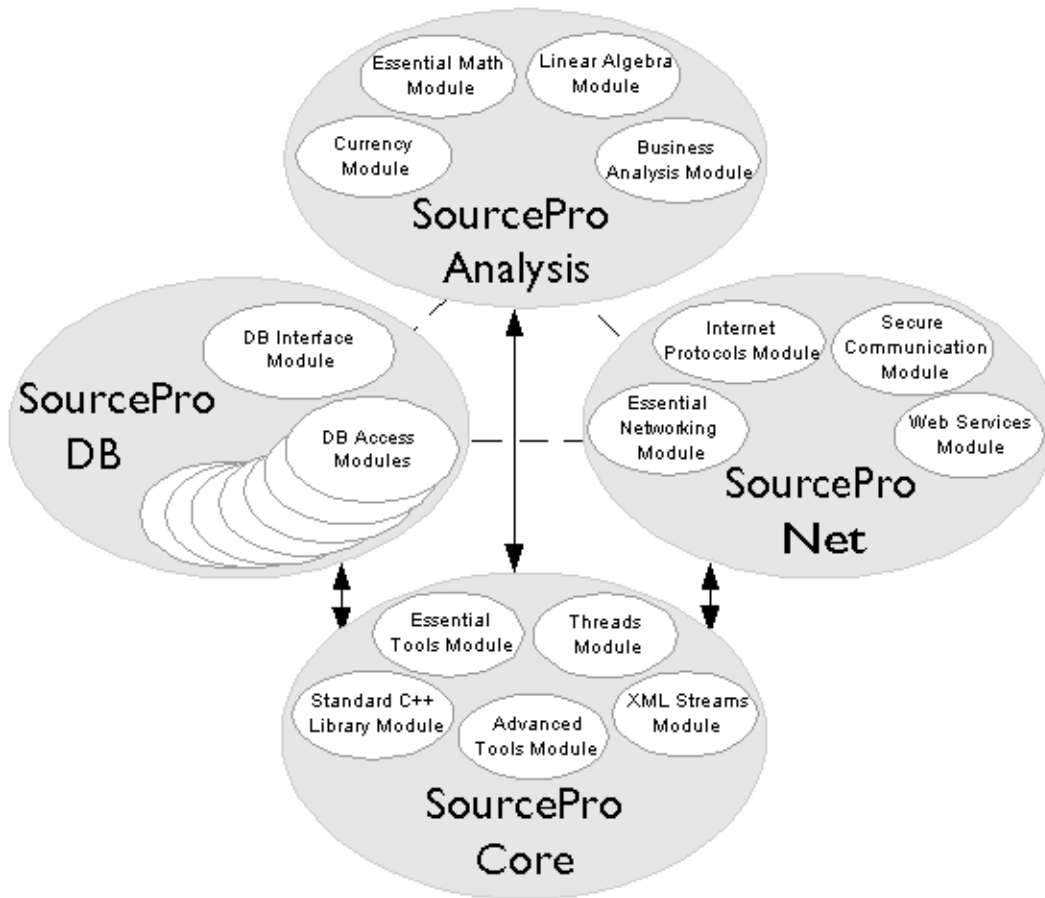


Figure 1 - SourcePro C++ Products and Modules

SourcePro Core

Rogue Wave SourcePro Core is our cornerstone product, containing an extensive set of fundamental C++ components that handle many of the intricacies of the C++ language. By letting SourcePro Core handle the granular details, developers can concentrate instead on solving the appropriate business logic challenges in their applications. Applications built with SourcePro Core can run on multiple operating systems, including Windows, Linux, and other popular UNIX operating systems, with minimal code changes. All SourcePro Core components are multithread safe, so developers can count on them to behave the same in a multithreaded environment or in a single-threaded environment.

SourcePro Core consists of five modules:

- Standard C++ Library Module
- Essential Tools Module
- Advanced Tools Module
- XML Streams Module
- Threads Module

Standard C++ Library Module

The Standard C++ Library Module is the Rogue Wave implementation of the International Standard for the C++ Programming Language, ratified by the American National Standards Institute (ANSI) and the International Standards Organization (ISO). Rogue Wave has played an active role in the ANSI/ISO committee for the standardization of C++, and leading vendors like Sun, Hewlett-Packard and Compaq ship our implementation of the ANSI Standard C++ Library with their C++ compilers.

The Standard C++ Library contains a broad range of general-purpose C++ functionality, but its low-level interfaces can be daunting and often error-prone. That's why SourcePro Core provides the Essential Tools Module, which enhances the Standard Library to provide richer interfaces, easier semantics, and extended functionality.

Standard C++ Library Module Features and Benefits

Data structures and algorithms. The Standard C++ Library Module provides many containers, including vector, list, set and map, plus a multitude of generic algorithms for performing operations on containers, such as initializing, searching, inserting and removing.

Locale facility. This extensible facility adapts the localization library to such conventions as numeric and monetary formats, character classification and order, and character code conversions.

iostream facility. This extensible facility provides a set of classes and templates that facilitate generic text input and output from or to device-independent streams. Iostreams use locales for internationalization.

Allocator class. Allocators provide a facility for the easy customization of the memory management policy used by individual containers.

String class. This class template provides string support to let developers easily manipulate narrow characters and other types (such as wide characters). Also included are facilities for substrings, copying, comparing, searching, inserting, removing, replacing, appending, swapping and more.

Numeric limits class. The numeric limits class provides an organized mechanism for describing the characteristics of the arithmetic types provided in the execution environment.

Complex class. This class template supports the creation of objects that represent and manipulate complex numbers.

Valarray class. The valarray class template defines an abstraction of a linear sequence of numeric values.

Essential Tools Module

The Essential Tools Module is an internationalized set of more than 130 C++ components that provide the basic building blocks for most C++ applications. The classes in the Essential Tools Module are useful in virtually any type of C++ application, offering consistent, intuitive APIs across classes to shorten the learning curve and help developers be productive quickly. With the Essential Tools Module, developers get a head start on laying the foundation for their C++ applications.

The Essential Tools Module is also designed to provide a less error-prone, easier-to-use interface to the underlying ANSI Standard C++ Library implementation, further insulating developers from the complexities of using the facilities of ANSI Standard C++ without any loss of integrity. Plus, the Essential Tools Module offers features beyond those in the ANSI Standard, including classes for handling dates and times.

Developers can create, manipulate and delete single, multibyte and wide-character strings with ease by using the Essential Tools Module. The module also provides a convenient framework for internationalizing C++ applications, allowing developers to write a single application that can ship to any country; when executed, the application will be able to process times, dates, strings and currency in the native format.

The Essential Tools Module provides multiple, extensive sets of collection classes, giving developers the flexibility to select the classes that best meet the specific needs of their applications. Finally, the Essential Tools Module supplies three kinds of extensible virtual streaming classes; developers can choose to maximize streaming efficiency, portability or both.

Essential Tools Module Features and Benefits

Complete set of string processing classes. These classes provide powerful support for creating, manipulating and deleting single, multibyte and wide-character strings. They also provide member functions for tasks such as reading, comparing and storing.

Multiple, extensive sets of collection classes, including:

- **RWCollectable Classes.** This complete library of collection classes is modeled after the Smalltalk-80 programming environment. These classes are particularly useful when developers need a simple programming interface, powerful I/O capabilities and high code reuse. These classes support simple, isomorphic and polymorphic persistence.
- **Template-based Collection Classes.** Some classes in this group supply a wrapper for each of the Standard C++ Library container classes, providing an interface with reference semantics. If developers have the Standard C++ Library installed, they can use the full interface to these classes; if not, a subset of these classes can still be used. Another set of classes in this group extend the Standard C++ Library containers with five classes written to match the standard. Each container offers value-based and pointer-based Rogue Wave wrapper templates.

Variety of extensible virtual stream classes. This set of virtual stream classes includes binary streams for when performance is critical but portability is not; portable streams for when portability is a higher priority than efficiency; and endian streams for a balance of portability and efficiency.

Convenient, easy-to-use framework for internationalizing C++ applications. With this convenient and easy-to-use eight-bit clean library, developers can write a single application and ship it to any country; when executed, the application will be able to process times, dates, strings and currency in the native format.

File system classes. These classes offer an easy way to add standard file operations and B-tree disk retrieval, and manage the free space in a file.

Error handling. The Essential Tools Module includes a complete error handling facility that takes advantage of C++ exceptions if they are available.

Advanced Tools Module

The Advanced Tools Module augments the Essential Tools Module in SourcePro Core, providing a range of C++ classes, global functions, macros and enumerator types for more power and flexibility in building C++ applications.

Using the streaming mechanisms in the Advanced Tools Module, developers have more flexibility than ever to move data into and out of their applications in a non-intrusive way. Flexible thread-safety options let developers control the level of thread safety for each stream in an application in order to maximize performance. Developers needing to stream data to and from internationalized applications will appreciate the powerful and flexible options for various stream formats and transformations.

The Advanced Tools Module's intuitive API makes it possible to stream nearly any C++ class with minimal code changes. To give developers even more options for serializing

objects, the streaming classes in the Advanced Tools Module can interoperate with the virtual streaming mechanism in the Essential Tools Module.

Advanced Tools Module Features and Benefits

Flexible non-intrusive streaming mechanism. The intuitive API in the Advanced Tools Module makes it possible to serialize nearly any C++ class with only minimal code changes. Developers have no need to derive from a base class to implement serialization. This provides a useful option for streaming data that ties in with the Essential Tools Module's virtual streaming mechanism.

Flexible thread-safety options. The Advanced Tools Module includes alternatives for setting the appropriate level of thread safety for each stream within an application. These important options allow developers to maximize performance by giving them control over the level of stream thread safety.

Flexible stream formats and transformations. Included in the Advanced Tools Module are streaming formats for narrow and wide characters (Unicode), as well as binary and data formats. The wide character and Unicode formats are ideal for moving data to and from internationalized character sets and platforms. Additionally, developers can quickly and easily create their own custom formats.

XML Streams Module

The XML Streams Module builds on the Advanced Tools Module, enabling C++ data to be written to or read from an XML stream. Developers can turn instances of C++ classes into XML without writing XML streaming code, making it easier to integrate existing C++ code into systems that use XML to communicate.

The XML Streams Module allows developers to serialize an object and write it as a stream of XML data. The serialized object can later be streamed in and restored, either by the process that streamed the object or by another process equipped to handle XML streams. The use of XML as the data format for object serialization has these advantages:

- The data stream is human readable.
- The data stream is self-describing, which allows for processing of the data by other XML-aware applications.

The XML Streams Module works directly with the data streams and object serialization classes of the Advanced Tools Module. Taken together, the classes of these two modules provide a comprehensive and flexible data streams architecture.

XML Streams Module Features and Benefits

XML streaming. The XML Streams Module gives developers a way to turn instances of C++ classes into XML without writing their own XML code. XML Streams serializes objects in XML format, making it easy to integrate existing C++ code into systems that use XML to communicate. And, when requirements change, new C++ code written today can interoperate with XML in the future.

Threads Module

The Threads Module offers a complete set of C++ components designed to help developers quickly build and deploy high-performance multithreaded applications. Although many of today's operating systems now provide support for multithreading and synchronization, this support is not consistent across platforms and must be accessed through low-level, procedural, C programming language APIs. The independent nature of these functions often lead developers to make implementation errors that may appear only at run-time. In addition, C++ developers prefer the object-oriented approach rather than the procedural style the native C threading libraries utilize.

The Threads Module provides a higher-level, object-oriented API that gives C++ developers a familiar, less error-prone way to work with the native C threading library implementations. The consistent threading API provided by the Threads Module on every supported platform makes multithreaded applications readily portable across Windows, Linux and other popular UNIX operating systems.

The fundamental classes for thread creation, control and synchronization in the Threads Module give developers the basics they need to create threads, manage threads and control access to shared resources. In addition, the Threads Module goes beyond the basic threading functions offered by the native C threading library implementations, providing many advantages to developers. Advanced threading abstractions like IOUs and producer/consumer queues help to improve the performance of multithreaded applications. The execution tracing facility simplifies the debugging process by providing an event-logging interface. Developers can use the smart pointer classes to more easily control dynamically allocated memory and reduce memory leaks in their application. Finally, the functor classes in Threads Module allow developers to retrofit existing single-threaded applications to create multithreaded applications, leveraging existing code to improve application performance without starting from scratch.

Threads Module Features and Benefits

C++ encapsulations of key multithreading concepts such as thread creation, control and synchronization. The higher-level object-oriented API insulates developers from the low-level complexities of dealing with the native C threading library implementations. Developers can control access to shared resources, increasing throughput and responsiveness in their applications.

Advanced threading abstractions, such as IOUs, server pools, and producer/consumer queues. The Threads Module contains prebuilt, fundamental threading classes that extend the basic threading functionality offered by the operating system. The implementations of proven multithreading design patterns give developers a head-start on building portable, multithreaded applications.

Efficient and flexible execution tracing facility. The Threads Module makes it easier for developers to identify the source of errors in applications, and lets them monitor a running system to diagnose problems.

Thread-safe and exception-safe smart pointer classes. These classes help developers manage memory and eliminate memory leaks in applications.

Functor classes for retrofitting existing single-threaded applications. The Threads Module provides functor classes to help developers quickly transform existing single-threaded applications into high-performance, multithreaded applications.

SourcePro DB

SourcePro DB provides object-oriented relational database access in C++. SourcePro DB has a layered architecture that abstracts away the complexity of writing database applications, yet allows developers to drill down to the native database client libraries if needed. SourcePro DB encapsulates ANSI SQL 92 and supplies a consistent, high-level C++ interface to relational databases, speeding development and reducing complexity.

SourcePro DB offers a layered architecture consisting of the database-independent Interface Module and a variety of database-specific Access Modules. This architecture allows developers to write programs that are reusable with the same or different types of databases, freeing developers from having to understand many of the details associated with a particular database vendor's API. SourcePro DB provides direct access to Oracle, Oracle 8, Sybase, DB2, Informix and MS SQL Server, as well as general access via ODBC.

SourcePro DB offers significant benefits for developers working with multiple databases as well as for developers working with a single database. For developers working with multiple databases, SourcePro DB lets developers write reusable programs for different databases with less effort than using a vendor API to write an application for a single database. SourcePro DB shortens the developer learning curve by encapsulating each database vendor's API; developers need to learn only a single interface since SourcePro DB provides a consistent object-oriented API across all supported vendor databases.

In an application that needs to access data from only a single database, developers can code to the database-independent SourcePro DB API for simplicity. This interface allows the application to work with virtually any back-end database in case project requirements change in the future. The object-oriented encapsulation of relational database concepts frees developers from having to generate SQL programmatically, while the intuitive C++ interface makes coding database applications dramatically less error-prone. Plus, SourcePro DB's cross-platform interface remains consistent even when vendors change their database API.

The OpenSQL API in SourcePro DB helps developers maximize performance on a specific database by providing lower-level access via a statement-based architecture, allowing developers to create their own SQL, bind the variables to the statement, and execute the statement. Overhead is minimized and the developer has direct access to native data types. In addition, SourcePro DB lets developers gain direct access to native structures and functions, allowing database-specific implementation choices to be made if the need arises.

SourcePro DB consists of two modules:

- Interface Module
- Access Modules

The two modules work together, with the individual Access Modules containing classes that implement Interface Module code for a particular vendor's database. Every

installation of SourcePro DB must contain both the Interface Module and an Access Module. However, Access Module classes are used only by the Interface Module to interact with the database; developers interact only with the Interface Module classes. This design lets developers write an application once, and reuse it—now or later—with different databases, simply by changing the Access Module.

The architecture of SourcePro DB is shown in Figure 2.

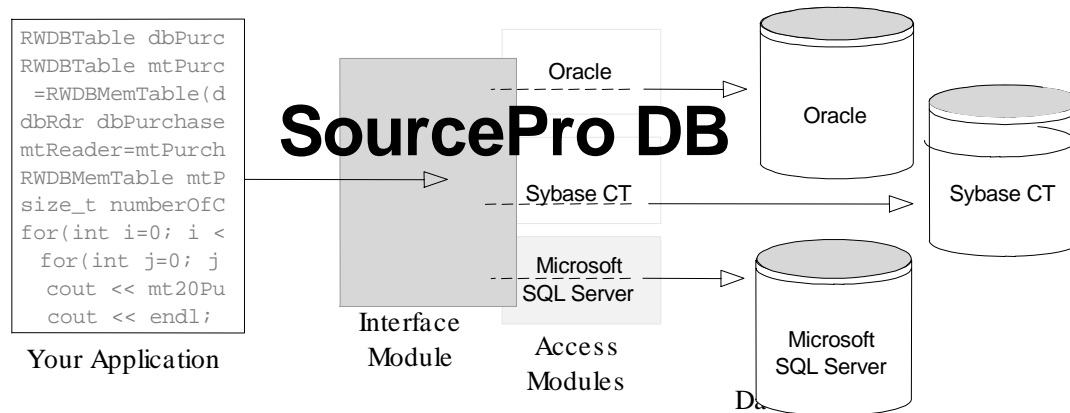


Figure 2 - The Architecture of SourcePro DB

Interface Module

The Interface Module provides a C++ programming interface that is invariable across all supported databases, and contains classes that encapsulate basic database constructs, such as connections and tables, and basic data manipulation operations, such as selecting, inserting, updating and deleting.

Interface Module Features and Benefits

Consistent, object-oriented programming interface to multiple backend databases. The Interface Module encapsulates leading relational database vendors' APIs and is universal for all applications regardless of which vendor's database is used. Developers need not deal with the low-level details of a particular database vendor's API, allowing them to concentrate on domain-specific issues and build appropriate data models. The Interface Module allows developers to drill down to the native database APIs for added control if needed, and frees developers from having to generate SQL programmatically.

C++ classes that encapsulate SQL 92 Data Manipulation Language (DML) and Data Definition Language (DDL) constructs. Developers can quickly learn to code multi-database applications because only one API must be learned. Higher-level abstractions allow for shorter, cleaner applications and facilitate code reuse.

Open SQL interface. Allows developers to take advantage of database-specific features to maximize performance, and to gain tight control over statements and their binding and execution semantics.

Optional schema caching. Can dramatically improve performance when used in situations where database schemas don't change often.

Access Modules

Each Access Module shares the features of the Interface Module, and provides native, high-performance implementations for supported vendor databases. This allows developers to write an application for one or more databases, and to later port to other databases with minimal code changes.

There are seven Access Modules, providing direct connectivity to leading databases and general access via ODBC:

- Access Module for DB2
- Access Module for Informix
- Access Module for MS SQL Server
- Access Module for Oracle
- Access Module for Oracle8
- Access Module for Sybase
- Access Module for ODBC

SourcePro Net

Rogue Wave SourcePro Net allows development teams to deliver object-oriented secure or non-secure networked and Internet-enabled applications with ease. SourcePro Net simplifies development by handling the granular details of secure socket programming and Internet protocols, helping developers deliver quality applications on schedule.

SourcePro Net lets developers use their existing C++ knowledge to implement standard SOAP concepts, in order to create Web Services and extend their existing investments in C++ through XML. SourcePro Net has a layered architecture, giving developers the flexibility to choose the level of abstraction right for their application: the most abstract layer for ease of use, or the protocol layer for fine control.

SourcePro Net consists of four modules:

- Essential Networking Module
- Internet Protocols Module
- Secure Communication Module
- Web Services Module

Essential Networking Module

The Essential Networking Module offers an intuitive, higher-level API for creating networking applications, allowing developers to be productive quickly without having to master the low-level details of socket programming. Developers who have used the de facto standard Berkeley socket API in the past will quickly master Essential Networking Module's C++ socket API.

The Essential Networking Module is designed with a flexible three-layer architecture. The low-level *Communication Adapter Layer* offers fine-grained control over the details of the socket protocol, while the *Portal Layer* provides a single interface to multiple platforms, network services and protocols. The high-level *Communication Services Layer* allows for transport-independent communication services, letting developers stream data across a network without having to know how the data is sent.

Essential Networking Module Features and Benefits

Clean, intuitive API for developing networking applications. The Essential Networking Module handles the messy details of socket programming and is easy to understand, so developers can quickly add networking capabilities and instead focus on the specifics of the application.

The Communication Adapter Layer encapsulates familiar network communication C APIs. Writing code to this lower-level API gives developers fine control over the features of a specific protocol. This layer includes a C++ Berkeley socket adapter that encapsulates this familiar TCP/IP interface with an easy-to-use C++ API.

The Portal Layer provides a single interface to multiple platforms, network services and protocols. Developers can write one set of Portal Layer code that can talk to any of the underlying communication adapters, simplifying development and ensuring portability.

The Communication Services Layer allows for transport-independent communication services. This high-level API allows developers to stream data into and out of network portals without knowing anything about the final output medium or formatting of the results.

Internet Protocols Module

The Internet Protocols Module provides a clean, intuitive API for developing Internet-enabled client-side applications, abstracting away the complexity of dealing directly with the granular details of key Internet protocols such as HTTP, SMTP, FTP and POP3. The API is easy to understand, so developers can be productive quickly without having to master the low-level details of the various protocols.

The Internet Protocols Module is designed with a two-layer architecture for flexibility and ease of use. The *Agent Layer* offers a simple, higher-level interface to key Internet protocols, allowing developers to perform complete protocol transactions with minimal code. The *Client Layer* provides control over the fine-grained details of a given protocol.

The Internet Protocols Module supports the HTTP 1.1 methods PUT, TRACE, DELETE, OPTIONS, GET, POST and HEAD. The module's HTTP 1.1 implementation also facilitates connections with multiple servers for improved performance and offers support for

persistent connections, reducing server overhead. Plus, it supports resumable downloading, enabling a file to continue downloading after a previous download has been interrupted. Finally, the HTTP 1.1 implementation helps developers create basic authorization, providing a fundamental element of security.

Internet Protocols Module Features and Benefits

Clean, intuitive API for developing Internet-enabled applications. The Internet Protocol Module is easy to understand, so developers can get productive quickly without having to master the low-level details of dealing directly with Internet protocols, such as HTTP, SMTP, FTP and POP3.

The Agent Layer provides a higher-level of abstraction to key Internet protocols. With minimal code, developers can quickly add HTTP, SMTP, FTP and POP3 transaction capabilities to their C++ client applications.

The Client Layer provide complete control over the low-level features of key Internet protocols. Developers can drill down for control over the fine-grained details of a protocol to build custom solutions for specific low-level protocol features that an application may require.

Support for the HTTP 1.1 specification. The Internet Protocols Module provides a C++ implementation of key areas of the latest HTTP specification, allowing developers to take advantage of specification improvements and create Internet applications conforming to the latest standards.

Secure Communication Module

The Secure Communication Module provides an easy-to-understand, intuitive API for developing secure Internet-ready applications. It contains C++ classes that use the Secure Sockets Layer and Transport Layer Security (SSL/TLS) and HTTPS protocols, which protect data that is being sent over the Internet.

The SSL/TLS protocols provide privacy and data integrity between two communicating applications by encrypting the data and by including a message integrity check. The HTTPS protocol is identical to the HTTP protocol except that it uses secure SSL/TLS sockets instead of using insecure TCP/IP sockets directly.

The classes in the Secure Communication Module abstract away the complexity of dealing directly with low-level security protocols so developers can be productive quickly. With the Secure Communication Module, developers have everything they need to equip their applications for encrypted low-level socket communication and encrypted HTTP connections.

The Secure Communication Module is based on proven, respected SSL implementations, including RSA's commercial product BSAFE SSL-C and the open source alternative OpenSSL, providing a higher-level, consistent, C++ interface to the security algorithms developers know and trust. Like that of the Essential Networking Module, the layered architecture of the Secure Communication Module gives developers the flexibility to use the higher-level API for simplicity or to drill down to gain fine-grained control over the details of the protocol if needed.

Secure Communication Module Features and Benefits

Clean, intuitive API for developing secure networking applications. The Secure Communication Module is easy to understand, so developers can get productive quickly without having to master the low-level details of dealing directly with SSL/TLS protocols.

Intuitive API for secure communication using HTTPS. The higher-level, intuitive API to the HTTPS protocol allows developers to write code more quickly by not having to deal with protocol level code. HTTPS classes derive from the Internet Protocol Module's HTTP classes and offer a consistent interface, so developers can get up to speed quickly. Developers can easily transform HTTP applications written with the Internet Protocols Module into HTTPS applications for secure communication over the Internet.

Consistent interface to industry-leading SSL implementations. The Secure Communication Module provides a consistent C++ interface to two leading SSL implementations: RSA's commercial product BSAFE SSL-C and the open source alternative OpenSSL.

Support for both encryption and authentication (X509 certificates). The Secure Communication Module includes everything developers need to quickly implement a simple secure connection between a client and a server.

Based on proven, respected SSL/TLS implementations. The classes in the Secure Communication Module offer a higher-level, consistent C++ interface to the leading implementations of security algorithms that developers know and trust.

Designed to easily plug into the architecture of Essential Networking Module. The architecture of the Secure Communication Module is intuitive because it mirrors that of the Essential Networking Module, providing an easy migration path for developers who have used the Essential Networking Module to build socket-based applications.

Web Services Module

The Web Services Module helps developers add XML support to their C++ applications, by means of Web Services, allowing applications to invoke business logic over the Internet using SOAP, the Simple Object Access Protocol. The Web Services Module is not tied to any single server or environment.

The Web Services Module simplifies the task of publishing business logic using SOAP by providing native C++ classes that represent the various SOAP entities in an intuitive and easy-to-use fashion. Using the Web Services Module makes it simple for developers to build applications that create, transmit and receive SOAP messages over the Internet or intranets using HTTP or HTTPS.

By providing an abstraction of the low-level SOAP details, the Web Services Module allows developers to handle SOAP documents in a familiar way using C++, reducing the learning curve and allowing projects to be completed faster.

Web Services Module Features and Benefits

Full encapsulation of the SOAP specification. The Web Services Module allows developers to convert SOAP objects to and from strings, enabling them to work with any networking or deployment technology.

Greater flexibility. The Web Services Module gives developers flexibility when working with SOAP by removing the need to hard-code SOAP representations into an application's business logic. SOAP serialization can take existing C++ code, and serialize/de-serialize the objects into SOAP payloads; there's no need to rewrite code to incorporate SOAP messages.

Allows SOAP to be transported over HTTPS. The Web Services Module works in conjunction with the HTTPS Package in the Secure Communication Module to transport SOAP messages over the Internet in a secure manner. This makes it easier than ever to code flexible business applications that incorporate new levels of privacy and security.

SourcePro Analysis

Rogue Wave SourcePro Analysis provides object-oriented classes for business and scientific analysis, offering a wide range of C++ components for solving mathematical problems in business and research. An intuitive, higher-level API shortens the developer learning curve and allows developers to devote more time to the business logic instead of focusing on the C++ implementation details.

SourcePro Analysis includes C++ classes for dealing with everything from basic math issues to specialized issues such as statistical analysis and currency conversion. In all cases, developers can rely on the algorithms in SourcePro Analysis for accurate, precise calculations, allowing them to instead focus on the business logic and build appropriate data models.

The intuitive C++ APIs in SourcePro Analysis provide an easier way to map mathematics to C++ by encapsulating the complexity of working with mathematical algorithms, shortening the learning curve and helping developers be productive quickly.

Code written using the consistent, intuitive APIs in SourcePro Analysis is more readable, more reliable, and more easily maintained.

SourcePro Analysis consists of four modules, each offering an intuitive C++ interface to essential mathematical algorithms:

- Essential Math Module
- Linear Algebra Module
- Business Analysis Module
- Currency Module

Essential Math Module

The Essential Math Module includes a broad range of basic math classes, including vectors, matrices and arrays, for developing robust numerical applications. It handles a full range of computational tasks, supplying classes for complex numbers, persistent streams, Fast Fourier Transforms, linear algebra decompositions, histograms, random distributions and linear regressions. Plus, the Essential Math Module has been optimized to provide the speed and efficiency of FORTRAN with the convenience and maintainability of an object-oriented language.

Essential Math Module Features and Benefits

Math collection classes, including vectors, matrices and arrays. The Essential Math Module dramatically simplifies programming for any code that manipulates arrays of numbers.

Classes for complex numbers, persistent streams, Fast Fourier Transforms, linear algebra decompositions, histograms, random distributions and linear regressions. The Essential Math Module's extensive set of classes makes it the ideal choice for financial modeling, signal processing, engineering applications, simulation modeling and more.

Optimized for speed. The Essential Math Module operates with the speed and efficiency that FORTRAN offers, while preserving the features that make C++ an ideal language for developing numerical applications.

Linear Algebra Module

The Linear Algebra Module offers a set of C++ classes that provide a complete, intuitive, object-oriented interface to the entities and algorithms of numerical linear algebra. The algorithms themselves are taken from LAPACK (Linear Algebra Package), a fast and efficient library of FORTRAN subroutines for solving common problems in numerical linear algebra.

Instead of having to master a procedural interface, developers can use the intuitive, object-oriented APIs to translate linear algebra problems from mathematics to C++ code—without sacrificing performance. The C++ interface allows developers to specify exactly which features they wish to control; other features automatically take on their default behavior. Because the objects in the Linear Algebra Module are as orthogonal as possible, developers need learn only about those objects relevant to their application. The flexibility and simplicity this provides are impossible to duplicate with a procedural interface.

The Linear Algebra Module expands the functionality of Essential Math Module by adding a wide range of specialized matrix classes for a total of 29 different matrix types, each of which has a similar interface, so learning is easy. It also includes factorization classes and classes for symmetric and non-symmetric eigenvalue decomposition.

Each of the classes in the Linear Algebra Module offers both simplicity and ease of learning for novice users. Advanced users enjoy the flexibility to control every aspect of the computation by choosing specific algorithms and algorithm parameters. The Linear Algebra Module offers a powerful cross-platform solution for development of financial analysis and scientific research applications.

Linear Algebra Module Features and Benefits

Intuitive C++ classes for performing numerical linear algebra. The Linear Algebra Module encapsulates the FORTRAN LAPACK library with intuitive, object-oriented APIs, delivering the power and performance of the well-tested public domain LAPACK algorithms without the hassles of using a procedural interface.

Specialized and sparse matrix classes. The Linear Algebra Module expands the functionality of the Essential Math Module by adding specialized matrix classes such as general dense, symmetric, skew-symmetric, Hermitian, banded, symmetric banded, Hermitian banded, upper and lower triangular and tridiagonal.

Factorization classes. The factorization classes in the Linear Algebra Module allow developers to perform simple operations without knowing about factorization, giving them more control over the details when needed for complex situations. Each of the factorization classes has a virtually identical interface, so developers need to only learn one type.

Symmetric and non-symmetric eigenvalue decomposition classes. The Linear Algebra Module encapsulates a single concept in each class, allowing developers to use only the objects needed to solve a specific problem. The sophisticated, carefully layered design makes it easy for even novice users to build an eigenvalue decomposition.

Other low-level, efficient decomposition classes. Included in the Linear Algebra Module are the basic building blocks developers need to handle more specific sorts of problems.

Business Analysis Module

The Business Analysis Module handles the granular details of statistical analysis so developers can concentrate on the business logic and create appropriate data models. The C++ classes are intuitive, object-oriented abstractions for performing sophisticated data analysis, allowing mathematical objects to be expressed using accepted arithmetic. To boost application performance, the classes in the Business Analysis Module take advantage of optimized BLAS (Basic Linear Algebra Subprogram) routines where available.

The Business Analysis Module offers a wide range of powerful and flexible functionality, including classes for performing multiple linear and logistic regression, parameter calculation and model selection. These classes let developers easily control synchronization of data, specify calculation methods and choose whether to work with supplied options for evaluating models or define their own.

Business Analysis Module Features and Benefits

C++ classes for performing sophisticated data analysis for business intelligence. The intuitive, object-oriented abstractions provided by the Business Analysis Module allow developers to use familiar arithmetic operators to express mathematical objects naturally. The Business Analysis Module takes care of the granular details of statistical analysis so developers can concentrate on domain-specific issues and on building appropriate data models.

Regression classes. The Business Analysis Module offers pre-built classes for both multiple linear regression and logistic regression. The regression classes couple regression data and parameter calculations so the currently calculated parameters and the regression data are always synchronized.

Parameter calculation classes. Developers can use one of the calculation methods provided in the Business Analysis Module to specify their own calculation method; developers can also change calculation methods at runtime.

Model selection classes. The Business Analysis Module provides multiple model selection techniques including forward, backward, stepwise and exhaustive selection, for both linear and logistic regression models. Developers can choose one of the provided model selection techniques or define their own model evaluation function.

Currency Module

The Currency Module supplies a variety of classes to make it easier for developers to deal with currency and money in C++ applications, including conversion policies, flexible formatting classes and classes for representing and manipulating decimal fractions exactly. Developers can use the transparent currency conversions and monetary calculations to specify the source of the exchange rates to be used, making it easier to use real live feeds or other sources of data.

The Currency Module provides default conversion algorithms that developers can extend or override in an application, according to business needs. While the C++ language's built-in floating point types only approximate decimal numbers, the Currency Module allows developers to represent up to 28 digits of precision. This helps protect mission-critical applications from expensive unreported overflow errors that can be caused by using built-in floating point types `float` and `double`. When using the precise arithmetic functionality, developers also have complete control over how to handle rounding errors in their applications.

The Currency Module provides the most complete solution to the euro challenge, combining currency conversions, monetary representations and decimal precision to help developers adapt existing or newly built systems to accommodate the euro. The Currency Module supplies conversion capabilities for both bilateral and the mandated triangulated conversion methods within the eurozone countries.

Currency Module Features and Benefits

Classes for representing currencies and money, a money calculator and conversion policies.

The Currency Module provides transparent currency conversions and monetary calculations, improving developer productivity. Developers can extend or override default conversion algorithms, allowing them to design an application that meets their specific business needs.

Currency conversion capabilities for both bilateral and triangulated conversion methods within the eurozone countries. Currency conversions, monetary representations and decimal precision help to give new or existing applications the capability of handling conversion of currencies to and from the euro.

Decimal classes for representing and manipulating decimal fractions exactly. The fixed decimal classes in the Currency Module supply automatic rounding in order to maintain

the correct number of decimal places. The Currency Module protects mission-critical applications from expensive errors by providing precise arithmetic as well as allowing developers complete control over how to handle rounding errors in their application.

Formatting class for formatted I/O. The Currency Module allows for the simple, highly flexible formatting of decimals, giving developers an easy way to add precise I/O formatting to their applications.

SourcePro C++ Products Working Together: A Code Example

This section provides a code example to demonstrate how Rogue Wave SourcePro C++ products work together smoothly to perform a given task.

In the following example, class *RWDBDatabase* from SourcePro DB and class *RWServerPool* from SourcePro Core are combined to create a higher-level abstraction, the multithreaded class *QueryServer*. In just a few lines of code, class *QueryServer* manages a controlled number of threads that can perform database queries for the whole application.

In the code using *QueryServer*, the server is constructed with an *RWDBDatabase* object, and the number of threads to be used at one time for database queries is specified. Following that, application code can submit queries to the *QueryServer* using its `enqueue()` method. Internally, *QueryServer* manages its threads using *RWServerPool*, and asynchronously gives results to users via the *RWTIOUResult* class template. When the application is done with the *QueryServer*, its destructor cleans up all the worker threads it created, and automatically frees up any used connections to the database.

QueryServer shows one way of easily wrapping up multithreaded database programming chores into an easy-to-use class for other programmers. However, the usage in `main()` does not exhibit its full capability. If 20 queries are submitted, the server would queue them and then send them to the database as connections are freed. This procedure allows developers to keep strict control over the access and the number of connections made to the database.

Example Using SourcePro Core and SourcePro DB

```
// Header file for class QueryServer
//
// From the SourcePro DB Interface Module
#include <rw/db/db.h>

// From the Threads Module
#include <rw/thread/RWRunnableIOUFunction.h>
#include <rw/thread/rwtMakeRunnableIOUFunction.h>
#include <rw/thread/RWServerPool.h>

class QueryServer
{
    RWDBDatabase database_;
    RWServerPool pool_;

    // Service function enqueued on the RWServerPool to handle
```

```

// the request. Private, because users should use enqueue()
// to place a query on the queue.
RWDBResult serviceFunction(const RWCString& sql)
{
    // Grab a connection, go for it.
    return database_.connection().executeSql(sql);
}

// Not defined on purpose
QueryServer(const QueryServer&);
QueryServer& operator=(const QueryServer&);

public:
QueryServer(const RWDBDatabase &db, unsigned concurrency)
    : database_(db), pool_(RWServerPool::make(concurrency))
{
    // Set the database's connection pool size to our
    // concurrency level for better performance.
    database_.defaultConnections(concurrency);

    // Start the pool running to accept requests.
    pool_.start();
}

~QueryServer()
{
    // Wait for the pool to finish processing its queue.
    pool_.stop();
    pool_.join();
}

RWTIOResult<RWDBResult> enqueue(const RWCString& sql)
{
    // Make a new runnable to handle the request and
    // throw it on the queue.
    RWTRunnableIOUFunction<RWDBResult> rf =
        rwtMakeRunnableIOUFunctionMAI(QueryServer,           // class
                                     *this,                 // instance
                                     RWDBResult,           // return type
                                     QueryServer::serviceFunction, // function
                                     const RWCString&,      // first arg type
                                     sql);                 // first arg

    pool_.enqueue(rf);
    return rf.result();
}
};

//
// Main program
//

#include <rw/db/db.h>
#include <iostream> // or use iostream.h for non-
                  // Standard C++ Library Module builds
#include "queryServer.h"
#include <rw/thread/RWRunnableSelf.h> // for rwSleep()

using namespace std; // use only for Standard C++
                   // Library Module builds

void outputStatus(const RWDBStatus& aStatus)
{
    if (aStatus.errorCode() == 0)
        return;

    // Print out the error.
    cout << "Error code: " << (int) aStatus.errorCode() << endl
    << "Error message " << aStatus.message() << endl
    << "Is terminal: " << (aStatus.isTerminal() ? "Yes" : "No")

```

```

        << endl
    << "Vendor error 1: " << aStatus.vendorError1() << endl
    << "Vendor error 2 : " << aStatus.vendorError2() << endl
    << "Vendor message 1: " << aStatus.vendorMessage1() << endl
    << "Vendor message 2: " << aStatus.vendorMessage2() << endl;
}

int main()
{
    RWDBManager::setErrorHandler(outputStatus);
    RWDBDatabase db = RWDBManager::database("ORACLE8",
                                           "database",
                                           "username",
                                           "password",
                                           "");

    RWDBTracer& trc= db.tracer();
    trc.setOn(RWDBTracer::SQL);
    trc.stream(cout);
    RWDBConnection conn = db.connection();

    // Create and populate table
    conn.executeSql("create table testServ(coll varchar(20))");
    conn.executeSql("insert into testServ values('Oregon Ducks')");

    // Create QueryServer
    QueryServer server(db, 5);
    RWCString query("select * from testServ");
    RTIOUResult<RWDBResult> resultIOU = server.enqueue(query);

    // Poll until redeemable
    while (!resultIOU.redeemable()) {
        cout << "Waiting for result..." << endl;
        ::rwSleep(50);
    }

    // Obtain RWDBResult object from RTIOUResult
    RWDBResult res = resultIOU.redeem();

    // Create RWDBReader
    RWDBReader rdr = res.table().reader();

    // Output data from query
    RWCString output;
    while(rdr()) {
        rdr >> output;
        cout << output << endl;
    }

    // Cleanup
    conn.executeSql("drop table testServ");
    return 0;
}

```

Related Publications

This white paper provides an overview of the range of functionality offered by the Rogue Wave SourcePro C++ products. More detailed technical information about each product can be found in individual white papers:

- [Rogue Wave SourcePro Core: Fundamentals for C++ Applications](#)
- [Direct Control of SQL in SourcePro DB Applications](#)
- [SourcePro DB: Offering Power and Productivity for C++ Database Applications](#)

- SourcePro Net: Solving Distributed Computing Problems
- Rogue Wave SourcePro Analysis: Solving Mathematical Problems in Business and Research
- XML- Enabling SourcePro DB Applications with SourcePro Net
- Enhancing SourcePro Core's Virtual Streams with the Advanced Tools Module

Solid Solutions from Rogue Wave Software

Putting Rogue Wave SourcePro C++ to work in developing mission-critical applications gives development teams an edge in achieving their schedule, cost and quality goals. SourcePro C++ provides essential components for fundamental programming, multithreading, platform-independent database access, networking and business and scientific analysis.

Applications built using SourcePro C++ are freed from specific hardware, operating system, and compiler limitations and insulated from dependence on a particular database API or vendor— all without sacrificing performance and control. SourcePro C++ gives applications the portability and adaptability to flex with inevitable changes, allowing existing investments in code to be preserved. Developers value the object-oriented APIs in SourcePro C++, knowing the built-in power and flexibility strengthens the reliability, performance and maintainability of their applications.

By facilitating more productive use of available internal technology resources and staff, SourcePro C++ helps development teams deliver reliable applications that solve business problems on time and on budget. SourcePro C++ helps development teams with these universal challenges:

- Short delivery schedules
- Increasing developer productivity
- Increasing application stability and maintainability
- Flexibility to modify applications to meet changing business requirements

Tens of thousands of Rogue Wave-powered mission-critical applications are currently deployed by Fortune 1000 companies in finance, telecom, software, aerospace and other industries. Companies building complex, high-performance applications can choose SourcePro C++ with confidence.



Corporate Headquarters
Toll-free: (800) 487-3217
E-mail: sales@roguewave.com

www.roguewave.com

The Netherlands

Rogue Wave Software B.V.
Telephone: +31 20 301 26 26

Germany

Rogue Wave Software GmbH
Telephone: +49 6103-59 34-0

France

Rogue Wave Software S.A.R.L.
Telephone: +33 1 41 96 26 26

United Kingdom

Rogue Wave Software U.K. Ltd.
Telephone: +44 118 9358600

Italy

Rogue Wave Software S.R.L.
Telephone: +39 02 4125.081

Japan

Rogue Wave Software Japan K.K.
Telephone: +81 3 3512-5012
E-mail: jpinfo@roguewave.com
www.roguewave.co.jp

Printed in USA
1-SP-WP1 9/01