

GemBuilder for Java: Providing Java connectivity using GemBuilder for Java

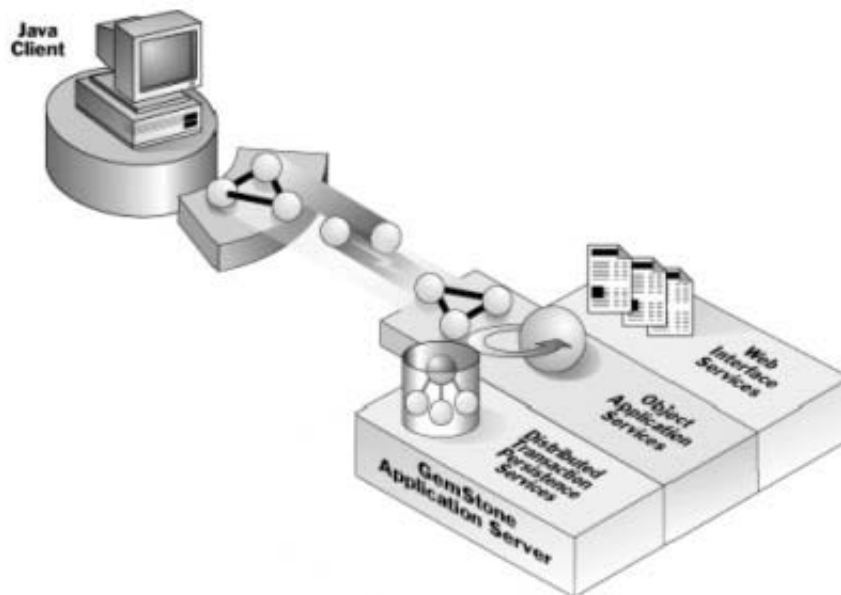
GemBuilder® for Java™ pairs the portability and flexibility of Java with the scalability of the GemStone/S™ Smalltalk object server for developing enterprise-wide business applications.

GemBuilder for Java lets you take advantage of the multi-user object execution engine, transaction management, and fault-tolerant environment of the GemStone/S object server.

GemBuilder for Java includes Java run-time classes that provide a Java client interface to GemStone/S business objects. The GemBuilder for Java development tools are implemented in Java, so developers can create Java clients and GemStone/S Smalltalk appli-

cations using their preferred Java development environment on any Java-capable workstation.

The GemStone/S object server is the key component in deploying Java applications across the enterprise. It gives Java clients fast, transactional, and concurrent access to business objects distributed across multiple platforms and databases. The GemStone/S object server supports the thin-client model through application partitioning. It scales up to thousands of concurrent users, providing concurrency control for large-scale distributed transactions as well as fault tolerance and recoverability for mission-critical applications.



Java client access to business objects

GemBuilder for Java gives Java clients direct access to large-scale, multi-user business object models running in the GemStone/S object server. With GemBuilder for Java, business applications can be partitioned so that user interfaces are handled by Java clients, while object and database queries and processing are handled by GemStone/S. Systems resulting from this model combine a full-featured, portable user interface with shared application logic and data—or business objects—on the server.

GemBuilder for Java provides the tools and classes to map Java client-side presentation objects to business objects in GemStone/S. It provides an efficient mechanism for invoking business object methods from Java applications. GemBuilder for Java's lightweight client API allows local Java messages to be forwarded to server-based GemStone/S objects. API classes facilitate connecting to GemStone/S, executing queries, creating and modifying server objects, and committing transactions. GemBuilder for Java performs brokered session management to minimize connection time and optimize performance.

Rapid development and deployment of distributed applications

GemBuilder for Java brings the portability and flexibility of Java to multi-user business applications. GemStone/S-enabled Java clients can be stored on a server and published to the end user through a Uniform Resource Locator (URL), rather than having to be installed on each machine in the organization. Java applets for accessing GemStone/S can be downloaded from any Web server to Java-enabled Web browsers and desktop systems; applications can thus be maintained in a central location; ensuring that users always receive the most recent version of the application.

GemBuilder for Java offers flexibility and ease of maintenance for business applications. Both Java and GemStone/S support dynamic binding; references to objects and methods are resolved at run-time, so objects and their interfaces can be updated as needed while applications are running.

Applications can check available services at run-time, taking advantage of new services as soon as they are available.

Industrial-strength Internet applications

GemBuilder for Java gives developers the ability to deliver responsive, large scale Web-based business systems built around Java run-time clients. Developers can take advantage of GemStone/S to create Java client applications with performance and scalability that can support thousands of concurrent users. Objects addressed through Java interfaces can access the millions of other objects in the business domain. And GemStone/S offers strict security at the object level, so Java applications can access selected public business objects, while critical business data is protected. GemStone/S provides fast, fine-grained access to the complex objects used for business modeling and for Web applications.

With GemBuilder for Java, business objects are stored in the transactional, fault-tolerant environment of the GemStone/S object server. Transactions are protected from hardware failure, and session state can be preserved through shutdown and startup. GemStone/S supports 24x7 operation with external administration.

Enterprise-wide information integration

Through GemStone/S' distributed architecture, Java applications can use objects residing anywhere in the enterprise. Through GemStone/S' connectivity tool, GemConnect®, Java applications can also access business data from legacy relational databases. In addition, Java applications can concurrently access and modify objects created in Smalltalk, while applications written in Smalltalk can access Java-created objects through the GemStone/S object server.

Developing Java applications for the enterprise

The GemBuilder for Java API is implemented in Java to maintain all the advantages of Java across platforms. To connect Java interface objects to GemStone/S business objects, a Java client calls the GemBuilder for Java run-time classes to start one or more GemStone/S sessions and initialize the interface with the GemStone/S object server. To access a particular

object, the client application uses the runtime classes to acquire a reference—a *stub*—to that object. The client then operates as if the business object were local, sending messages to the stub, which forwards the messages to the appropriate server object for execution, returning results to the Java client.

GemBuilder for Java tools

The GemBuilder for Java tools are written in Java, integrating seamlessly into the Java IDE of your choice. This set of tools provides client developers access to all GemStone/S business objects as well as GemStone/S extensive built-in class library. Includes:

- **System Browser:** The Browser is used to view the available GemStone/S classes and methods, create new classes, and add or modify methods.
- **Inspector:** The Inspector gives developers a view of business objects residing in the GemStone/S object server. The developer can examine the state of individual objects, modify them if desired, browse collections, or look at the contents of dictionaries.
- **Workspace:** The Workspace allows developers to execute arbitrary strings of GemStone/S code. This allows developers to locate business objects to examine in an Inspector, create sample data for testing applications, assign access control to objects or collections, and perform administration tasks in the GemStone/S object server.
- **Debugger:** When execution of a GemStone/S method results in a run-time error, the Debugger is available, allowing developers to review the execution stack, looking for logical errors or objects that contain unexpected state.



Corporate Headquarters:

1260 NW Waterhouse Ave., Suite 200 Beaverton, OR 97006 | Phone: 503.533.3000 | Fax: 503.629.8556 | info@gemstone.com | www.gemstone.com

Copyright© 2007 by GemStone Systems, Inc. All rights reserved. GemStone® and the GemStone logo are trademarks or registered trademarks of GemStone Systems, Inc. Information in this document is subject to change without notice.