

Progress® DataXtend™ CE for C#

Full Lifecycle Data Services for High Performance .NET Applications

Highlights

Progress DataXtend CE for C# provides cost-effective, scalable, and high performance object-oriented access to relational data for Microsoft® .NET applications. DataXtend CE meets and exceeds data access requirements at each stage of the project lifecycle, ensuring a fast, free flow of data.

Features at a Glance

Professional Edition:

- Graphical object-relational mapping tools that integrate with Visual Studio® .NET, and model-driven, interactive code generation to accelerate development.

Enterprise Edition Adds:

- Built-in intelligent cache that understands the object model and schema for high performance.
- Transparent database optimizations for SQL Server™, as well as all major databases, for native performance and vendor independence.
- Cache clustering for scalability and high availability.

Think Data First!

Plan for data access early in a custom development project, or accept the risk that data bottlenecks will plague application performance and limit scalability.

Introduction

Performance, availability, and scalability concerns often take a back seat to application functionality until the end of a development project. However, in *Improving .NET Application Performance and Scalability*,¹ the authors warn: “Many, if not most, performance problems are introduced by specific architecture, design, and technology choices that you make very early in the development cycle, often in the design stage.” The strategies they recommend for building performance into an application stress the importance of choosing a scalable solution for persistent data access.

Analysts also confirm that data access technology has a significant impact throughout the project lifecycle:

- **Development Phase:** An R.B. Webber study concluded that coding and configuring data access typically accounts for 30 to 40% of total project effort.
- **Tuning Phase:** Performance and data integrity issues often do not surface until testing.
- **Deployment Phase:** According to a 2004 Forrester Research survey, 66% of performance problems are first identified by irate users calling the IT help desk.

Progress® DataXtend™ CE for C# provides cost effective, scalable, and high performance object oriented access to relational data for Microsoft® .NET applications. DataXtend CE meets and exceeds data access requirements at each stage of the project lifecycle, ensuring a fast, free flow of data. Tightly integrated features shorten development schedules, simplify performance tuning, and ensure enterprisewide deployment scalability and high availability.

In the previously referenced document, a Microsoft architect strongly suggests that development teams plan up front for performance: “If you’re very lucky, performance problems can be fixed after the fact. But, as often as not, it will take a great deal of effort to get your code to where it needs to be for acceptable performance. This is a very bad trap to fall into.

“At its worst, you’ll be faced with a memorable and sometimes job-ending quote: “This will never work. You’re going to have to start all over.””

When you use DataXtend CE’s development tools, you build runtime performance and scalability into your application. You gain the flexibility to seamlessly access sophisticated tuning and deployment features without rearchitecting or additional coding.

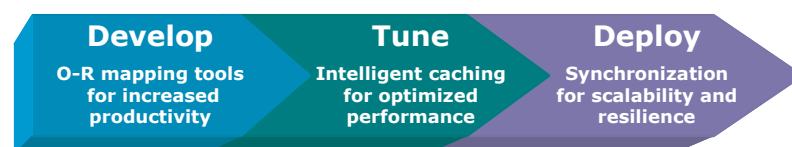


Figure 1. DataXtend CE optimizes data access across the project lifecycle.

Beyond the initial cost savings, DataXtend CE's data services platform reduces the risk of project failure due to schedule overruns, poor performance, or application failure. With dozens of large-scale deployments at global Fortune 500 companies, Progress Real Time Division products enjoy proven success in the most demanding enterprise systems.

The DataXtend CE Advantage

Enterprise architects have long promoted an object-oriented approach to relational data access. Representing relational data as objects provides a level of abstraction that relieves developers from low-level database details and from embedding query language in their code. It allows developers to encapsulate data access, rather than replicating it in many different locations and creating a maintenance nightmare.

Object-relational (O-R) mapping has been time-tested for C++ and Java application development. Effective mapping boosts developer productivity and improves application architecture. It allows developers to work with application objects and to specify through a tool or configuration how those objects map to database tables and columns.

Unfortunately, the primary API for data access in the .NET framework, ADO, does not take an object oriented approach. ADO.NET exposes a relatively low-level API for manipulating rowsets. Although ADO code can be generated either through Visual Studio or by third-party tools, the generated code requires extensive modification to optimize data access for demanding applications.

In contrast, DataXtend CE for C# offers a time-tested, flexible, full-lifecycle, and cross-platform approach that meets data access requirements from development through deployment. Professional and Enterprise editions offer a choice in feature sets and a seamless upgrade path. Both provide development features. The Enterprise edition adds tuning and deployment features:

– **Development features:** Available in both the Professional and Enterprise editions. Tools for rich O-R mapping and model-driven code generation speed design and implementation. Database schema browsing and import automates specification of the application data model.

- **Tuning features:** Available in the Enterprise edition for high performance. Adds intelligent caching for the generated mapping classes to dramatically reduce the load on centralized data sources. Configurable cache management policies optimize performance with no additional coding.
- **Deployment features:** Available in the Enterprise edition to provide cost-effective scalability and resilience for deployment within the data center and beyond. Clustering, cache synchronization and high availability features keep your application up and running even under peak loads. When user demand increases, these features support incremental infrastructure expansion without code changes.

DataXtend CE's flexible and extensible data access platform is optimized for demanding applications exploiting new architectural patterns, especially those with complex data models, high request rates, or both. This comprehensive data access solution eliminates data bottlenecks that limit performance and scalability or even contribute to project failure.

Development Features Increase Productivity and Reduce Risks

During the development phase, meeting functional specifications and delivering the application as quickly as possible take top priority. It is simply inefficient for developers to spend a lot of time writing, testing, and debugging data persistence logic — plumbing code that should “just work” without a lot of extra effort. Instead, developers need to focus their efforts on the unique business requirements of their application — while using a persistent data infrastructure that builds performance and scalability into their application.

DataXtend CE for C# Professional and Enterprise editions meet these key requirements in three ways:

1. By simplifying data model design and O-R mapping with automated tools.
2. By generating a high quality, flexible data access layer that reduces testing and debugging efforts.

Documented Development Savings

Progress Real Time Division customers have compared the efforts involved in hand-coding data access with automatic data layer generation:

- In one case, every hand-coded persistent class contained about 1,000 lines of code. Not only did use of DataXtend CE-generated code result in a five times increase in productivity, it also reduced bugs by 60%.
- In another case, defects were reduced by more than 90%. The total savings for an application with several million lines of code was estimated at nearly two million dollars when factoring in reduced development, testing, and maintenance efforts.

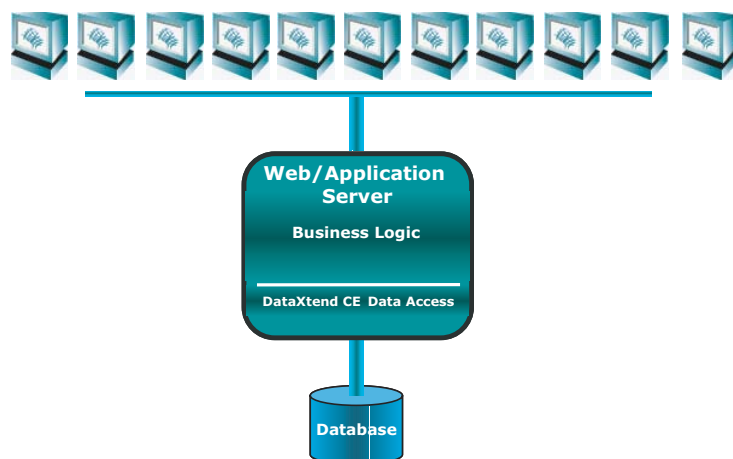


Figure 2. A DataXtend CE data access layer in a Web server or application server provides O/R mapping, and optionally caching, and cache synchronization.

3. By integrating the generated code with intelligent caching to ensure performance and data integrity without additional programming. (Enterprise edition only)

Progress Real Time Division's patented O-R mapping and caching technology reflects the sophistication and reliability of hundreds of man-years of research and development — delivering battle-tested code for the data access layer. The reduction of development, testing, and maintenance delivers quantifiable cost savings as documented by our customers.

Flexible Design Tools

DataXtend CE O-R mapping tools offer a choice of interface. Each of the tools define a project that captures O-R mapping and code generation options. In addition to the generated code, they produce a Visual Studio project file that simplifies integration into a new or existing solution. You can easily move DataXtend CE projects from one tool to the other.

DataXtend CE Workbench Tools provide multiple views of your application data model. A Database Explorer view allows you to browse existing database schemas and drag-and-drop schema elements into your project. Many O-R mapping tools map existing classes to new tables, but provide only limited support for existing tables. DataXtend Workbench Tools allow designers to start from either the application data model or from the database schema.

For those who prefer UML modeling, DataXtend CE Add-Ins for Rose provides O-R mapping and code generation for IBM®'s Rational Rose®. The DataXtend CE Object Builder offers another alternative interface.

Benefits of a Data Services Layer

The persistent data access code generated by the DataXtend CE tools abstracts the data source, allowing you to switch the database without changing application code or recompiling. By supporting not only SQL Server™, but all major relational databases, use of DataXtend CE prevents database vendor lock-in and simplifies migration. Yet, the underlying data access code takes advantage of each database's unique performance features.

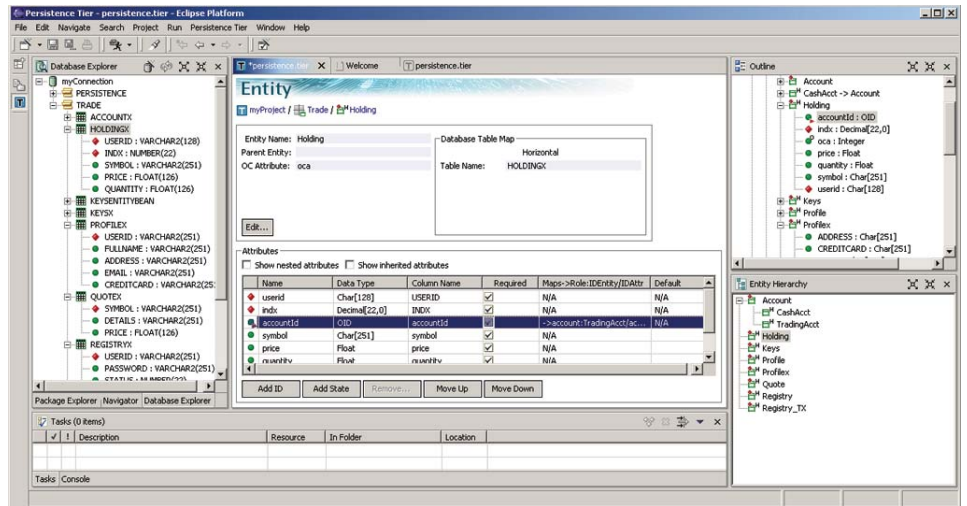


Figure 3. DataXtend CE Workbench Tools simplify data layer development and increase productivity. The Database Explorer (long pane on left) provides drag-and-drop convenience for applications that will use existing tables.

DataXtend CE tools can generate C#, Java, and C++ code, providing a consistent or enterprise-wide data layer across services in a service-oriented architecture. DataXtend CE can integrate with Microsoft .NET, BEA® WebLogic®, and IBM WebSphere® application servers. No other O-R mapping tool offers such complete language and platform support.

Tuning Features Ensure Data Access Performance

Simply getting an application to run is only half of the battle. Meeting response times and transaction rate requirements during load testing is equally important for project success. At this stage of the project lifecycle, decision-makers tend to throw bigger hardware at the performance problem.

While hardware has become more affordable, hidden costs include personnel for maintenance and the development time required to distribute the application — not to mention additional database licenses and operational expenses. And, disproportionate additions of computing power may be required to achieve adequate performance.

The Microsoft resource "Improving .NET Application Performance and Scalability," identifies caching as an important strategy for the data access layer of an application. However, they warn against caching data that changes frequently or that is distributed in multiple processes because of the additional programming, testing, and maintenance required to ensure data integrity.

DataXtend CE for C# Enterprise edition removes this restriction with intelligent caching that was designed for distributed applications with complex data models and information that changes dynamically. DataXtend CE for C# Professional edition allows you to develop an application and take advantage of DataXtend CE's performance and scalability when needed. By upgrading to the Enterprise edition, you can use intelligent caching and synchronization without rewriting your application.

Built-In Intelligent Caching

DataXtend CE's intelligent caching offers several key features to support applications with complex data models and/or high request rates:

- **Code-free cache management:** DataXtend CE's object-relational mapping classes are integrated with the cache — objects are automatically instantiated in the cache when the application accesses them. Developers do not have to manually track which objects must be cached, when they should be cached, or when they should be cleared from the cache.
- **Data abstraction:** The intelligent cache knows both the application data model and the database schema. This ensures that DataXtend CE materializes exactly one cache object for each row that it loads from the database. Developers do not need intimate knowledge of the database schema nor do they need to actively manage data integrity. They can take advantage of object-oriented benefits such as inheritance and encapsulation.

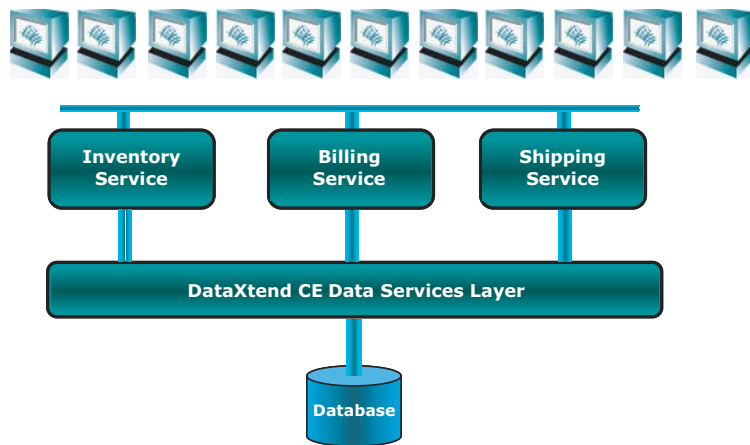


Figure 4. A DataXtend CE enterprise data services layer provides high performance and scalability in a service oriented architecture.

- **Safe, concurrent access:** A multi-threaded application can simultaneously read and/or update cached objects. DataXtend CE ensures data integrity and avoids supplying stale versions of the object to the application's threads. Applications can serve more client requests with fewer database connections.
- **Dynamic relationship caching:** Intelligent caching remembers not just the values of object attributes, but also the relationships between objects. Unlike other caching products, DataXtend CE can dynamically instantiate relationships, or it can pre-populate the cache with frequently-used relationships. By caching relationships, DataXtend CE can dramatically reduce the number of expensive join queries performed by the database, resulting in a corresponding improvement in performance.

While most applications perform well with the intelligent cache's default settings, every application has its own data access patterns. DataXtend CE's advanced cache management service and database tuning features (such as cache indexing, lazy loading, and database write policies) are configurable and do not require code changes or recompilation.

Customers have dramatically reduced the frequency of database access in their applications. Depending on the balance of read versus write operations, response times improve anywhere from 50–1000%. Customers typically see several-fold improvement in the number of concurrent users that can be served using the same hardware.

Continuous Cache Coordination

Object-oriented applications built on the advanced caching capabilities of DataXtend CE operate in complex environments where other enterprise applications that make use of traditional SQL access methods may be modifying the data as well. It is critical that updates made by these other enterprise applications are reflected in the object cache in a timely and consistent way.

Typical O/R mapping and caching infrastructures follow a traditional "request-response" model for accessing data. In this model changes made by other enterprise applications would not be seen until an explicit request is made to reload the cached data. This can result in accessing stale or even inconsistent data.

To address the caching requirements of these complex environments, DataXtend CE features Continuous Cache Coordination to proactively "push" changes made to the database out to the distributed cache. This ensures that the enterprise data caching infrastructure maintains fresh and

consistent data. The advanced data caching technology of Continuous Cache Coordination is only available with DataXtend CE.

Deployment Features Deliver Data Access Scalability and High Availability

Application success depends on the scalability and resilience of the production system. Ironically, business success can trigger application failure when unexpected demand overwhelms the system. Sluggish response times can doom an otherwise elegant application, as will incorrect data. Similarly, without fault tolerance, the data access layer is much more brittle, allowing even minor network or database outages to threaten application availability.

For the most demanding, dataintensive applications, DataXtend CE's deployment features ensure that the performance and scalability achieved during pre-production continues through rollout — within the data center and beyond. Applications based on DataXtend CE's data services platform can achieve cost-effective scalability and reliability with cache clustering. DataXtend CE for C# allows you to take full advantage of .NET clustering features and complements them with stateful cache synchronization, guaranteeing fresh data for all users and supporting load balancing and failover implementations.

Scalable, Resilient Cache Synchronization

Cache synchronization uses messaging to ensure that the distributed caches remain up-to-date. DataXtend CE cache synchronization offers a choice of delivery modes: express and guaranteed. Express synchronization sends the synchronization

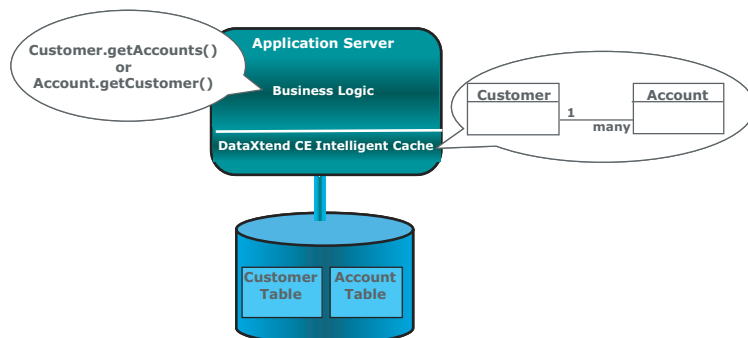


Figure 5. The intelligent cache understands the database schema and the application data model, simplifying data access while ensuring data integrity.

message immediately after a transaction completes when speed is the highest priority. Guaranteed synchronization integrates the sending of the synchronization message with the transaction and also writes the message to a persistent store, protecting against data loss even in the face of multiple system failures.

Because DataXtend CE's synchronization requires only basic configuration, servers can be brought online at any time to handle load increases or spikes, future proofing your system. DataXtend CE also provides automatic reconnection to the database, insulating users from a common cause of application failures and increasing uptime.

Cache synchronization also provides key benefits for advanced deployments involving service-oriented architectures or remote data centers:

- No IT department can effectively build custom SOAs without first implementing a shared data layer that guarantees data consistency across all services. Yet many use adhoc data access solutions that are unable to support enterprisewide data consistency. Progress Real Time Division provides data access infrastructure software specifically designed to provide consistent performance and highly available data across distributed computing environments.
- The advanced synchronization capabilities within DataXtend CE allow cost-effective deployment to multiple geographic locations. A "virtual data center" architecture requires only one central physical data store and uses replicated caching to enable applications to be deployed to remote locations. By eliminating the need to replicate expensive databases and the personnel and hardware they require, virtual data centers dramatically reduce infrastructure and maintenance costs, while ensuring data consistency.

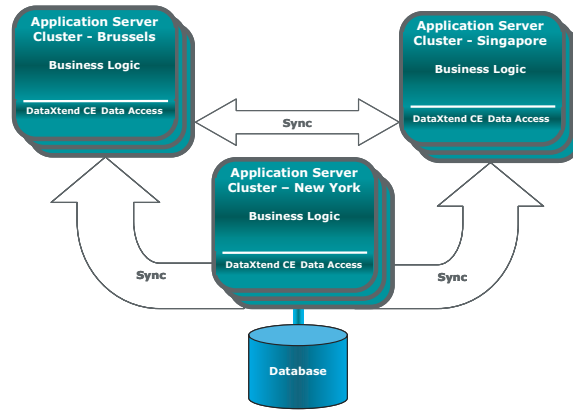


Figure 6. Virtual data centers use cache synchronization to ensure data consistency and improve local response times.

Services to Ensure Success

Our world-class Customer Care organization offers tiered support and consulting options that cover you from application design through deployment — even in situations when you might encounter problems from interactions with other vendors' products. Progress Real Time Division's customer-focused offerings include:

- Dedicated Technical Account Managers to assist every customer through successful deployment.
- Intensive online and classroom training that brings developers up to speed quickly and ensures that they take full advantage of DataXtend CE's capabilities.
- Expert services to provide advanced troubleshooting in situations where other vendors' tools may be affecting your application performance.
- An experienced Professional Services team offers architectural assessment, design and implementation assistance for even faster deployment.

The Leader in Data Access Infrastructure

Progress Real Time Division is the only open and cross-platform technology provider of data access infrastructure. Our intelligent caching infrastructure and cache synchronization patents extend those ground-breaking patents in data access technology.

Major partners like Adobe, Sun, Cisco and Sybase, have validated Progress Real Time Division's leadership by licensing our state-of-the-art technology, and our award-winning software has been recognized in *Intelligent Enterprise's* "IT Dozen," *Java Report's* "Great Java Products," and the *Java Developer's Journal* "Editor's Choice Awards."

More than 100 Global 2000 customers rely on Progress Real Time Division products to provide data access for the world's most demanding applications including real-time trading systems, transactionintensive logistics and package delivery systems and complex network management systems. To experience DataXtend CE for C#'s developer productivity, performance, scalability and availability for yourself, visit www.progress.com/realtime for a free 30-day evaluation.

DataXtend CE for C# Specifications

Development Platforms

Windows® XP, 2000, 2003

Visual Studio .NET 2003

Suggested Development Configuration:

Processor: PIII or greater

RAM: 256 MB or greater

Disk space: 165 MB

Deployment Platforms

Windows® XP, 2000, 2003

Database Support

SQL Server™

Oracle®

Informix®

Sybase®

MySQL®

IBM® DB2

DataXtend CE for C# Features

New in Version 3

Database Explorer	Included with the Workbench Tools in both the Professional and Enterprise editions, supports connecting to a database, browsing schemas, and importing schema elements into an DataXtend CE project to further increase developer productivity
Monitoring APIs	Supports creation of routines to monitor runtime execution in the Enterprise edition, simplifying tuning
Auto-configuring synchronization	For TCP messaging in the Enterprise edition, allows clustered servers to find each other on a local network, simplifying deployment
Automatic connection recovery	Releases and re-establishes dropped database connections within a pool, insulating application code from errors
MySQL® support	Supports this popular database in both the Professional and Enterprise editions, providing more options for deployment

Professional and Enterprise Edition Development Features for Productivity: Object-Relational Mapping

Tools include Plug-Ins for Eclipse, Add-Ins for Rose, and Object Builder	Support a variety of development processes to increase productivity, giving you the choice to work with an IDE, a modeling tool, or a simple standalone interface to define your DataXtend CE project.
Database Explorer	Included with the Workbench Tools, supports connecting to a database, browsing schemas, and importing schema elements into a DataXtend CE project to further increase developer productivity
Primitive data types	Maps object attributes to database column types, eliminating the need to write conversion routines
Composite data types	Combines multiple columns into a single object attribute structure, simplifying access to logically related information
Inheritance	Supports union and horizontal mapping, optimizing application performance for different database schemas
Cardinality constraints	Captures the semantic properties of related objects, protecting database integrity
Multi-attribute primary keys	Uniquely identifies objects with a combination of attributes, preserving object identity without changing the database schema
Deletion propagation constraints	Models the existence properties of relationships, preserving data integrity
Character set support	Transparently translates character strings between application and database, simplifying development of international interfaces
Database-independent APIs	Abstracts underlying database from application logic, allowing transparent switching of underlying database without re-coding
Database-specific optimizations	Automatically exploits database-specific features when possible, increasing overall performance
Stored procedures	Enables compliance with administrator-imposed data access rules, avoiding intrusive disruption
Optimistic concurrency	Automatically tracks object modifications, protecting data integrity while supporting concurrent reads for application-level scalability
Object locking	Manages concurrent access to shared data with row-level locks or version attributes, enabling fast concurrent access to data without compromising integrity
Transactional cache	Isolates object modifications within a transaction until commit, preserving ACID properties while supporting concurrent access for performance and scalability

Enterprise Edition Tuning Features for Performance: Intelligent Cache

Object-model aware	Understands both the application object model and the database schema, preserving object uniqueness
Relationship aware	Caches relationships, dramatically shortening response times
Multi-class fetch	Pre-loads cache with objects and relationships, improving efficiency, increasing cache hit rates, and decreasing redundant database queries
Indexing	Builds an in-memory index for non-key attributes, speeding retrieval of cached objects
Eager and lazy object loading	Supports retrieval from large result sets as needed, reducing network round trips
Eager and lazy relationship loading	Allows related objects to be loaded on initial access or as needed, providing flexibility for improved performance
Clearing policy	Provides explicit lifecycle management, clearing unneeded objects to improve cache efficiency while ensuring that frequently-requested objects remain to increase cache hit rates
Transient objects	Exposes all cache features, providing consistent infrastructure for application data that will not be written to persistent storage
Monitoring APIs	Supports creation of routines to monitor runtime execution, simplifying tuning

Enterprise Edition Tuning Features for Performance: Database Optimizations

Connection pooling	Automatically manages shared connections, reducing the load on the database
Array reads	Exploits database-specific optimizations, decreasing network traffic and increasing performance
Hints	Allows application to specify query hints, optimizing performance for complex queries
Write policies	Sends changes to database immediately or only at commit time, providing the best performance for each circumstance
Lazy attribute loading	Loads large attributes like BLOB and CLOB as needed, reducing network traffic and memory consumption
Batch updates	Sends multiple updates in one database request, reducing network round trips and increasing efficiency
Sparse updates	Sends only modified attributes to the database, improving performance for objects with many attributes
Partitioning	Allows object data to be distributed across multiple databases, supporting easy integration with existing data

Enterprise Edition Deployment Features for Scalability and Resilience: Cache Synchronization for Clustered Servers

.NET server support	Integrates seamlessly with .NET clustering features, leveraging your investment
State-based updates	Transmits object state instead of simply invalidating cached objects, supporting caching of dynamic data and increasing scalability
Express messaging	Ensures cache consistency without writing to disk, maximizing performance (Supports Tibco Rendezvous and self-configuring TCP messaging)
Guaranteed messaging	Temporarily writes synchronization messages to persistent store, ensuring cache integrity across application or network failures (Supports Oracle AQ)
Auto-configuring synchronization	For TCP messaging in the Enterprise edition, allows clustered servers to find each other on a local network, simplifying deployment

Enterprise Edition Deployment Features for Scalability and Resilience: Cache Synchronization for Clustered Servers (continued)

Open API	Allows replacement of synchronization transport, simplifying integration with corporate messaging standards
Cross-platform support	Enables synchronization across applications written in different language and deployed on different platforms, providing the only enterprise solution for heterogeneous environments

Enterprise Edition Deployment Features for High Availability

Failover	Complements application server's failover capability with synchronization, ensuring that cached data is consistent and increasing uptime
Load balancing	Keeps the cached data in clustered applications synchronized, allowing requests to be correctly handled by any server
Automatic connection recovery	Releases and re-establishes dropped database connections within a pool, insulating application code from errors
Manual database reconnect	Detects failures, closes and reopens all connections in the pool, accelerating recovery from intermittent database outages

¹From the Microsoft® patterns & practices series, by J.D. Meier, Srinath Vasireddy, Ashish Babbar, and Alex Mackman, April 2004.

PROGRESS
SOFTWARE

www.progress.com

Specifications subject to change without notice.

© 2005 Progress Software Corporation.

All Rights Reserved.

EXJ/BC/0705-

Code 2483



0000105333

Worldwide and North American Headquarters

Progress Real Time Division, 14 Oak Park, Bedford, MA 01730 USA Tel: +1 781 280 4000

UK and Northern Ireland

Progress Real Time Division, 210 Bath Road, Slough, Berkshire, SL1 3XE England Tel: +44 1753 216 300

Central Europe

Progress Real Time Division, Konrad-Adenauer-Str. 13, 50996 Köln, Germany Tel: +49 6171 981 127

France

Progress Real Time Division, 3 Place de Saverne, Les Renardières B, 92901 Paris la Défense Tel: +33 1 41 16 16 56

www.progress.com/realtime

© 2005 Progress Software Corporation. Progress and DataTend are trademarks or registered trademarks of Progress Software Corporation in the U.S. and other countries. Java and all Java-based marks are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries. Any other trademarks or service marks contained herein are the property of their respective owners.