

“We were looking for a high-performance embedded solution for persistence in a Java environment – Perst was by far the hands-down winner among the products that we tested.”

– iSpheres

### Overview

Perst is a high performance **embedded object-oriented database** for **Java** and **C#**. Perst is tightly integrated with the Java and C# programming languages. In contrast to object/relational databases, or tools that provide object/relational mapping, Perst stores data directly in Java/C# objects. This eliminates the need for expensive (in performance terms) run-time conversions between Perst's representation of the data and the Java/C# representation.

Unlike many other object-oriented databases, Perst requires no dedicated compiler or pre-processor, yet provides a high degree of application transparency. The Perst API is convenient, flexible and easy to use. And Perst is very compact. The engine's core is just 5,000 lines of code; depending on database access patterns, the run-time requires just 30K to 300K of RAM. Perst requires no end-user administration, and despite its simplicity, Perst ensures integrity via transactions that adhere to the ACID properties (Atomicity, Consistency, Isolation and Durability) with very fast recovery.

Perst is ideal for Java and C# applications requiring a modest footprint and fast, multi-platform data management. Typical uses include packaged software, mobile and embedded applications, Web services and industrial systems.

### Perst programming interfaces

Perst's fundamental achievement is to make working with persistent Java or C# objects as **efficient** and **easy** as possible. In most cases, Perst automatically loads the persistent objects without explicit programmer command. When used with aspect-oriented programming (AOP) tools such as AspectJ, Perst provides completely transparent persistence. Perst also implements a subset of SQL for filtering elements of a collection. For efficient access to persistent objects, Perst also implements specialized collection classes optimized for different data layouts and access patterns. These include:

- A classic B-Tree implementation that reduces disk I/O for general purpose data access;
- For spatial data representation, R-tree indexes (based on Guttman's algorithm) that aggregate objects into proximity-based groups and then use proximity to further aggregate the groups;

- Main-memory database containers, based on T-Tree indexes, optimized for memory-only access;
- The Patricia Trie index to support IP address data;
- A TimeSeries class to efficiently deal with small fixed-size objects (stock quotes, for example);
- Specialized versions of collections for thick indices (indices with many duplicates), and bit indices (keys with a restricted number of possible values).

### Perst options

In addition to the core functionality, Perst provides optional features such as garbage collection, detection of hanging references, automatic schema evolution, XML import/export utilities, master-slave replication support (with the option to run read-only queries on slave nodes), an SQL subset to filter elements of any collection, and integration with the widely-used, open source AspectJ and JAssist AOP tools.

### Perst host platforms

Perst is available for Java and C# hosts. Perst for Java supports J2SE/J2EE versions 1.3 and 1.4, as well as J2SE/J2EE version 5. The Perst for Java platform version 5 supports all new features of the JDK 1.5 including template collection classes (generics), enums, and iterable classes.

The C# (Perst.NET) implementation supports C# versions 1.0 and 2.0 with the same source code. Support for specific C# 2.0 features (such as template classes) is provided at compile time.

Perst for C# includes a larger set of primitives than the Java implementation: C# supports unsigned integers, properties, enums, decimal type and structures. In all other aspects, the two Perst implementations are functionally equivalent with the same API.

Perst.NET supports standard and compact .NET frameworks, enabling Perst's use in Windows CE and Pocket PC applications.

#### McObject LLC

22525 SE 64th Place Phone: +1 425 888 8505  
Suite 302 Fax: +1 425 888 8508  
Issaquah, WA 98027 www.mcobject.com

**Precision Data Management**

