

Perst Lite for Java J2ME platform

McObject's Perst Lite is a micro-footprint version of the Perst open source, object-oriented embedded database. Perst Lite targets embedded systems and intelligent devices developed on the Java 2 Platform, Micro Edition (J2ME), bringing object-oriented database services to the embedded Java development community.

Perst Lite aims for a simplicity of design despite what McObject describes as barriers to object-oriented databases inherent in J2ME. Perst Lite also achieves a code size that is about 30% smaller than standard Perst, well within the resource constraints of most intelligent embedded devices.

Embedded systems developers increasingly use Java, especially as microprocessors improve in their speed executing Java code. Sun Microsystems' J2ME, with its flexible user interfaces, robust security, and built-in networking, is proving popular for consumer electronics and mobile devices.

But J2ME differs in major respects from standard Java (Java 2 Platform, Standard Edition, or J2SE), complicating vendors' ability to provide popular Java application development

Perst is an object-oriented embedded database for Java and .NET applications that need to deal with persistent data. It is easy to use and provides high performance. Its main advantage is its tight integration with the Java and C# programming languages. There is no gap between the database and the application data model since Perst stores data directly in language objects. Thus, there is no need for data-packing or unpacking code to map between the application's data model and the database's data model, in contrast to the requirements of relational and object-relational databases. Unlike many other object-oriented database management systems, the Perst embedded database requires no special compiler or pre-processor.

tools for embedded Java. One of Perst Lite's major achievements is to substitute its own lightweight code for standard Java's reflection mechanism.

Object-oriented database systems for Java rely on the language's reflection capability to inspect the object format at run-time and discover the format of the stored object. But J2ME, which targets small-footprint and embedded systems, omits reflection due to this function's complexity and resource requirements. This has limited the availability of OODBMSs for J2ME.

Perst Lite surmounts this obstacle by using a utility that generates object serialization and de-serialization code for the application that incorporates the database. This code replaces re-

flexion, but without the complexity and large run-time code footprint that caused reflection to be dropped from J2ME.

Another programming concept, 'weak references,' is commonly used by Java OODBMSs to permit garbage collection (automatic reclaiming of memory) while preventing out-of-control object cache growth. However, because weak references complicate garbage collection, many J2ME variants for specific devices do not support them. To address this issue, and permit efficient memory management, Perst Lite incorporates alternative object cache logic that does not depend on weak references.

Many developers prefer to use object-oriented database systems when working in object-

oriented languages such as Java, crediting object database systems with better programming language integration than their relational counterparts. When a relational database is used, the application needs code that translates the relational representation of the data to the object representation required by the host language.

Perst Lite retains most of the features popular with the thousands of developer who have downloaded the open source Perst since 2003: B-tree, Patricia Trie, Bit index, T-Tree and R-Tree indexes as well as List, Relation, and Set collections, all protected by transactions supporting the atomicity, consistency, isolation and durability (ACID) properties. Perst Lite also offers multithreaded access, data encryption and asynchronous replication.

Under Perst's dual license, users can modify Perst and Perst Lite database source code and use it freely in other open source applications (software for which source code is made available) under the GNU General Public License (GPL). McObject's commercial license is required if source code of the Perst- or Perst Lite-based applications is to be withheld.

Subsystem combines audio and MPEG-4 video

ARC International's ARC Player Subsystem is the latest addition to its range of programmable subsystems that are suitable for system-on-chips (SoCs) targeting media-intensive consumer applications.

The ARC Player Subsystem provides an MPEG-4 video decoder and multi-standard audio codecs coupled with a configurable 32-bit microprocessor as a pre-verified solution. It is available for licensing by qualified customers globally.

Increasingly consumer electronics manufacturers are incorporating multimedia functionality into a range of devices, such as portable audio players, electronically enabled toys, and low-end cellular phones.

This trend is causing semiconductor companies to migrate from the 8- and 16-bit microcontrollers and fixed hardware blocks that are prevalent in these products today, to new cost effective 32-bit microprocessor-

based subsystems, which offer multi-standard audio, MPEG-4 video capabilities, and much greater computing performance.

The ARC Player Subsystem meets the needs of these applications by integrating audio and video onto a programmable subsystem that is based upon a 32-bit CPU/DSP core.

Components included in the pre-verified ARC Player Subsystem are a member of the configurable 32-bit ARC 600 core family, MPEG-4 video decoder, MP3 decoder, audio/video synchronization, and voice recorder. Optional audio decoders include WMA, AAC, and Dolby Digital Consumer (AC-3).

A set of software and hardware development tools also are available, simplifying the SoC design process and speeding customers' ARC-Based silicon implementations to market.

Featured in the ARC Player Subsystem

is a member of the configurable 32-bit ARC 600 core family. It is a full-capable embedded core that can serve as a control, applications, and DSP processor using a unified RISC architecture. ARC Player is also a general-purpose 32-bit processor with the power to handle the communications functions required in consumer applications.

Because the ARC Player Subsystem is programmable, SoC designers can include additional codecs or algorithms to extend the market life and reduce the need for future product development.

The ARC Player Subsystem can be evaluated using the ARCangel 4 (AA4) demonstration platform, which also serves as a highly integrated development system. The AA4 incorporates an FPGA-based ARC Player Subsystem running the open-source Linux operating system.