

Alternative Technologies  
**Relational Access Manager (RAM) Standard Product**

**PRODUCT DESCRIPTION**

Relational Access Manager Standard Product is a shared library that maximizes developer productivity, relational database management, and relational database efficiency. Designed by experts in relational database design and applications, it is highly portable and supports a number of computer systems and relational databases. RAM provides:

- \* C language interface
- \* support of Oracle, ShareBase (Britton Lee), and Sybase
- \* optimized access to relational data
- \* automatic conversion between data structures
- \* isolation from database complexity and change

**BENEFITS:**

- \* flexibility in database schema design
- \* database vendor independence
- \* programmer productivity
- \* database administrator control
- \* reduction of software maintenance costs
- \* reduction of training costs

**GENERAL FEATURES:**

RAM is a shared library written in C and running under UNIX System V and VAX/VMS.

**Freedom to change your database design without recoding your applications is yours with RAM.**

Alternative Technologies (AT) offers RELATIONAL ACCESS MANAGER as a significant advance in database applications development. It was designed by experts with experience in relational database design and one-line transaction processing (OLTP) applications development.

RELATIONAL ACCESS MANAGER was designed to support applications such as on-line computer intergrated manufacturing and fixed income portfolio management. Flexibility requirements for these applications resulted in the broad utility and benefits of RAM.

The RELATIONAL ACCESS MANAGER shared library has many capabilities. Among others, it provides a smart interface to the relational database, executing multiple SQL statements with a single function call. In doing so, it isolates both the SQL and relational database schemas from C application code. It allows experienced C programmers, SQL coders, and relational database administrators each to concentrate on the portion of an application they understand best.

The RAM runs under UNIX System V and VAX/VMS.

The benefits of the RAM Standard Product include:

- \* ease of programming
- \* programmer productivity
- \* database administrator productivity
- \* personnel management is improved
  - C programmers do not have to know SQL
  - SQL and database 'gurus' do not have to know C or other development languages
- \* object-oriented interface
- \* improved datatype support
- \* improved per task performance
  - bind and parse time reduced
- \* flexibility
  - data dictionary not bound to application

- \* improved maintenance
  - code and database queries are cleanly separated
  - changes in data dictionary or database queries do not force recompiling or relinking
  - RAM code can be a shared library or a server so all processes share it
- \* applications are smaller and more uniform
- \* improved system performance
  - smaller tasks mean fewer page faults
- \* vendor database independent code is encouraged

If the cost of database maintenance is high enough, changes to the schema will be forbidden in order to avoid that cost, whether it be time, expertise, or potential disruption of the business. This coupling between application code and database schema effectively removes on the primary benefits of a relational database - it is supposed to be flexible.

The RAM is designed to eliminate these problems. It is available in two forms: as a library (the standard product) and as a server (the extended product). The standard product provides a number of functions which can be called of the C programming language, and, in certain environments, from other languages as well. These functions isolate the third generation language code from the relational database language code. They provide a standard interface for the programmer to use in accessing the relational database. Furthermore, they isolate vendor specific relational features from the application so that relational database management systems vendors can be changed without modifying third generation language code.

All this is done without sacrificing performance. The size of applications is minimized by reducing redundant database access code. The code is extremely portable across environments. In effect, the intended flexibility of relational databases is not only preserved, but extended to the application code.

The cost of maintenance is decreased, debugging time is reduced, and neither the application programmer nor the database administrator need not be concerned about coupling between the application and the relational database. Each can do that portion of the work which they know best.

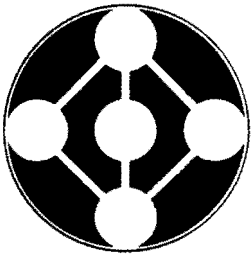
The major emphasis of the design of the RAM is to satisfy the repeatedly stated requirement that application programs be able to handle many different types of data structures and multiple DML statements as a unit. In order to achieve this aim, the concepts of object oriented programming, in particular "data abstraction", have been used extensively.

The RAM has been written so as to maximize the efficiency and simplicity of database access and updates (set-at-a-time and non-procedural) as requested by applications (single record-at-a-time and procedural). It makes extensive use of the data dictionary so that changes to the database do not affect its integrity.

The use of scripts, DML commands, stored commands, stored programs, and stored procedures as may be supported by a specific vendor, all provide a "call-by-name" syntax. For some database products, the RAM eliminates constraints on what kind of database language statement can be processed. For products which do not as yet support stored procedures, scripts become "virtual stored procedures" which can be created and maintained independently of either the application or the database code and reside on either the host file system or in the database.

In summary, the Relational Access Manager:

- \* isolates database specific code,
- \* hides the details of error processing,
- \* provides an easy method of manipulating data,
- \* improves application developer productivity, and
- \* improves system-wide performance.



Alternative Technologies  
**Relational Access Manager Extended Product**

**PRODUCT DESCRIPTION**

The Relational Access Manager (RAM) Extended Product is a software server version of the Relational Access Manager. It builds on the functional capabilities of the RAM shared library. Users of RAM can upgrade their environments to the Extended Product without rewriting existing applications. The RAM Extended Product was designed to provide transaction processing management in multiple-application relational database environments. The Extended Product provides a number of features which are not available through the shared library:

- \* a software server to manage database access
- \* a Transaction Definition Language
- \* an error recovery facility

**BENEFITS:**

- \* improved system performance
  - transaction manager removes contention between tasks before it happens
  - smaller tasks mean better memory management
- \* improved personnel management
  - an application system database administrator can control transaction management separately from application code design and development
- \* improved application maintenance
  - applications are simpler and easier to understand
- \* task prioritization
- \* task-level transaction management
- \* uniformly managed system-wide shutdown
- \* automatic deadlock recovery
- \* time-out and asynchronous interrupt support

**GENERAL FEATURES:**

The Relational Access Manager Extended Product is a software server running under UNIX System V and VAX/VMS.

**Now you can manage transaction processing in a relational database environment.**

Alternative Technologies (AT) offers RELATIONAL ACCESS MANAGER EXTENDED PRODUCT as a significant advance in database applications transaction management and recovery. It was designed by experts experienced in relational database design and OLTP applications development.

The RELATIONAL ACCESS MANAGER Extended Product which provides access to the library via a server process with various extensions such as global transaction management and significantly more robust error recovery.

Centralized control of database access provides a means for performance optimization and for transaction management at the application level. Here we distinguish between database transaction as handled by the relational database vendor and higher level relationships between transactions called application transactions. Application level transaction management serves to optimize concurrency, enforce serialization as required, and provide system-wide recovery and soft-failover.

Integrating OLTP and relational databases is a serious problem. The goal of the database vendor is to safeguard the consistency of the database. Unfortunately, this may have one serious drawback: it introduces unnecessary processing delays. It can effectively forbid concurrency and multiprogramming and therefore would degrade such important aspects of the performance of the multiple database application system as response time and transaction throughput. To ensure integrity, the program developed in isolation must bracket each of its transactions. This results in a loss of concurrent database access.

The Relational Access Manager Extended Product Transaction Definition Language provides the database administrator means for defining potential conflicts between transactions. If a potential for run-time conflict can be pre-determined, the Extended Product automatically removes the conflict by staggering the conflicting transactions. Where transaction requests from concurrent applications result in reading or writing the same entities of the database, the Extended Product can bracket database steps which are potentially in conflict. Transactions can then be dynamically and intelligently managed rather than enforcing a policy of safety-first. RAM Extended Product improves system throughput and response time. It also handles recovery from catastrophic program failure by backing out transactions.

Three facilities are part of the RAM Extended Product. The first is a transaction definition language. Its function is to allow the program developer to define the database steps, the entities upon which they operate, their required sequence, and critical transaction steps in such a manner as to drive the operation of the transaction manager. The second is a means for analyzing database transactions in order to perform the specification in the transaction definition language. The third is the heart of the manager itself; an interpreter of the transaction definition language capable of detecting conflicts.