VB/ISAM - CB/ISAM

The "un-database": easy for beginners, powerful for pros,
tiny, rock-solid, and *THE FASTEST YOU CAN GET.*

BASICPro (Now VB Programmer's Journal) READER'S CHOICE AWARD

Indexed-Sequential Access Method
extensions to:

Visual Basic for Windows
Visual Basic for MS-DOS
C/C++ (Windows)

Software Source

42808 Christy St., Ste. 222
Fremont, California 94538 USA
Tel +1-510-623-7854
Fax +1-510-651-6039

****

PRODUCT SUMMARY (VB/ISAM for Windows)

--> Low-level, complete data management engine (record manager); automatically maintains up to 80 sorted indexes per file.
--> Known as the fastest data manager available for VB/Win.
--> Uniquely compact (low-overhead) DLL -- under 50 KB.
--> Flexible: allows variable-length fields and records, huge records.
--> Single-user and multi-user/network versions, all file-compatible, for: VB/Win; VB/DOS; C/C++ (Windows).
--> Designed for ease of use: only 16 functions (most apps use only 6 or 8); 68-page user manual.
--> Bug-free and data-protective; proven since August 1991.
--> Inexpensive.

****

Data organization is at the heart of real computer programs, but Visual Basic and C don't give you a way to find John Smith's data without searching the entire file. VB/ISAM and CB/ISAM let you instantly find data through automatically-entered cross-references -- up to 80 per record.

Why the "un-database"? Because VB/ISAM and CB/ISAM give you powerful, automated indexed data management without the difficulties and overhead of a complicated database management system. In Visual Basic, for instance, most apps use only six or eight (of the 16) VB/ISAM functions, and the manual is under 70 pages long. Overhead is tiny -- the VB/ISAM-for-Windows DLL is under 50K!

MAJOR FUNCTIONS:

VmxCreate	VmxPut	VmxBOF	VmxGet/Lookup
VmxOpen	VmxDelete	VmxEOF	VmxGet/Next
VmxClose			VmxGet/Previous

HERE'S ALL YOU DO: First, call the VmxCreate function to describe the format of the data records in your file, and identify their index fields. Then, as you add, rewrite, and delete records, VB/ISAM - CB/ISAM immediately updates all the file's indexes, maintaining each of them in perfect sort sequence. Each index works like a smart Rolodex file, with direct lookup positioning and next/previous access. Our users have told us that VB/ISAM for Windows is by far ***THE FASTEST DATA MANAGER YOU CAN BUY FOR VB/WIN.***

****

HERE'S WHAT OUR USERS SAY:

"Stupendously fast."
"Very quick, very flexible, very complete"
"Fast, simple, reasonably priced, and lean."
"Bulletproof."
"The manual is even fun to read."
"The best-kept secret in Visual Basic!"

****

POWERFUL AND AUTOMATIC: VmxGet/Lookup sets a pointer for a specified index to the insertion point of your lookup value, even if it didn't find an exact "hit"-and you can step through the index from that point to get all Smiths, all part numbers from J22 to L14, etc. VmxBOF takes you to the beginning of an index for full-file sequential access in that index's sort sequence; each index provides a different "virtual sorting" of the entire file, performed ahead of time!

--> Up to 80 indexes per file, immediately updated in sort order when you add, modify, and delete records
--> Built-in optional comparison tests for next/previous loop control: equals, begins-with, etc.
--> All indexes stored with data records in one DOS file
--> File/index space is automatically maintained; no periodic reorganization/compaction required
--> Multiple files can be open at the same time

FLEXIBLE DATA ARCHITECTURE:

--> Variable-length records and fields: any/every record field can be a variable-length string field of unlimited size, up to max record size of 64 KB
--> Variable-length-string index fields to 250 bytes
--> Mixed record types in the same file (different substructures after common index fields)
--> Sparse index capability (not every record need be represented in every index)

EASY TO LEARN AND USE:

--> Only eight main functions-VmxCreate, Open, Close, Get, Put, Delete, BOF, EOF
--> 68-page (VB/ISAM for Windows) small-format manual, praised for its friendly and readable style; with sample code

FAST, SOLID, AND PRACTICAL: VB/ISAM•CB/ ISAM design evolved from 20 years of database software development for critical industrial applications and widely-distributed information systems.

--> Proven in commercial applications since 8/91
--> Exceptional speed with files up to 512 MB
--> DLL size (VB/ISAM for Windows) under 50 KB
--> All functions thoroughly test for parameter, record, and file validity, and return status codes
--> VmxWriteNote, ReadNote provide named file annotation strings for descriptions, version, etc.
--> Multi-user/network (shared update) versions work on any network, and include semaphores for record, group, and file locking; locks are automatically cleared if a user "dies"
--> File-compatible single- and multi-user versions for VB/Windows, VB/DOS, C/Windows
--> Unlimited distribution license -- no royalties

SUMMARY OF FUNCTIONS (VB/ISAM FOR WINDOWS)

All functions return integer status codes with symbolic equivalents defined as Global Constants: 0=OK, 1=NOT_FOUND, etc.

"Primary keys" are unique record identifiers-for example, Social Security Numbers for employee records. Secondary index keys, taken from the contents of designated string fields in the data records, need not be unique (so you can have multiple "Smith"s, etc.).

A VB/ISAM - CB/ISAM "file" consists of a single DOS file containing all indexes together with all data records, plus two smaller auxiliary files.

VmxCreate (FileName, MaxPrimaryKeyLen, TuningOptions, RecordFormatDescriptionString)

... Creates a new VB/ISAM - CB/ISAM file. The Record Format Description string specifies record layout and which string fields are index fields (with their maximum expected lengths, up to 250 bytes). For example, if the record layout for this file is defined by the following:

```
Type EmployeeRecordType
  LastName          As String      'Index, max 60 bytes
  FirstName         As String
  MonthlySalary     As Integer
  LotsOfText        As String      'Could be huge
  DepartmentCode    As String*5    'Index, 5 bytes
  MastersThesis     As String      'Really huge
End Type
```

...then the corresponding VB/ISAM record format string is "X60$, $, %, $, X$*5, $".

VmxOpen (FileName, AccessMode, FileNum)

... Opens a VB/ISAM file for either read-only or read-write access, returning a number for all future references to this open file.

VmxClose (FileNum)

... Closes an open VB/ISAM file.

VmxPut (FileNum, PrimaryKey, RecordVariable, UpdateMode)

... Adds or rewrites a record. RecordVariable names a structure (Type/End-Type) variable; it and its Primary Key are the source of the data. UpdateMode can be 1="add only", 2="replace only", or 0="don't care".

VmxDelete (FileNum, PrimaryKey)

... Deletes a record from a VB/ISAM file.

VmxGet (FileNum, Index, **XLOOKUP**, LookupString, PrimaryKey, RecordVariable)

... In Lookup mode, VmxGet directly positions a pointer in the specified index of the specified file to the "insertion point" of your LookupString. If that insertion point contains an entry (i.e., if the lookup found a match), the function also returns the corresponding record's primary key and (in the RecordVariable) data fields. Even if the lookup doesn't find a match, you're still properly positioned for a follow-up series of VmxGet/Next or VmxGet/Previous accesses.

VmxGet (FileNum, Index, **XNEXT**, CompareOption, OptionalComparisonString, IndexEntry, PrimaryKey, RecordVariable)

... In Next mode, VmxGet steps forward from the current position to the next entry in the specified index of the specified file, returning the value of that index entry and the corresponding record's primary key and data fields. If you use this function within a loop (e.g., to find all "Smith"s, all part numbers beginning with "K3", etc.), you can optionally have it return the result of comparing the encountered index entry against a parameter string (in several ways- equals, begins-with, less-than, etc.) as a convenient way to tell when to stop looping (name no longer equals "Smith", part number no longer begins with "K3", etc.).

VmxGet (... **XPREVIOUS** or **XCURRENT** ...)

... Similar to VmxGet/Next, for back-stepping or re-read.

VmxBOF, **VmxEOF** (FileNum, Index)

... Reposition the pointer of the specified index of the specified file to the beginning or end of that index.

OTHER FUNCTIONS:

VmxEncode and VmxDecode are unique data conversion functions for packing substructures and arrays into and out of records. VmxWriteNote and VmxReadNote allow you to add, rewrite, delete, and read named file annotation strings. VmxFlush does a quick save-your-changes-to-disk; VmxInfo returns file statistics; VmxReturnCode$ translates numeric function return codes into descriptive strings; and VmxKill deletes a VB/ISAM - CB/ISAM file.

In addition, the "MU" (Multi-User) products include VmxLock and VmxUnlock functions to set and clear semaphore strings for network record/group/file locking.

PRICE LIST (as of September 1, 1993)

(Please order by full product name:)

VB/ISAM MX for Windows, V2.3	(For Visual Basic for Windows -- single user)
VB/ISAM MU for Windows, V2.3	(For Visual Basic for Windows -- multi-user/network)
CB/ISAM MX for Windows, V2.0	(For C/C++ for Windows -- single-user)
CB/ISAM MU for Windows, V2.0	(For C/C++ for Windows -- multi-user/network)
VB/ISAM MX for DOS, V2.0	(For Visual Basic for MS-DOS -- single-user)
VB/ISAM MU for DOS, V2.0	(For Visual Basic for MS-DOS -- multi-user/network)

All products are royalty-free.

All prices are in U.S. dollars.

$30 discount for factory-direct orders ***by VISA, MasterCard, or prepaid check only:***
$ 99.95 FOR ANY MX (SINGLE-USER) PRODUCT (plus shipping/handling)
$169.95 FOR ANY MU (MULTI-USER/NETWORK) PRODUCT (plus s/h)

Invoice/list/retail prices:
MX: $129.95 (plus s/h)
MU: $199.95 (plus s/h)

Shipping/handling charges (California addresses, first add 7.25% tax):
US orders (2-day priority): $ 8.00
Foreign orders (airmail): $15.00
FedEx or COD: CALL.
