FairCom DB

v13

DB Made Simple

FairCom

DB v13

Table of Contents

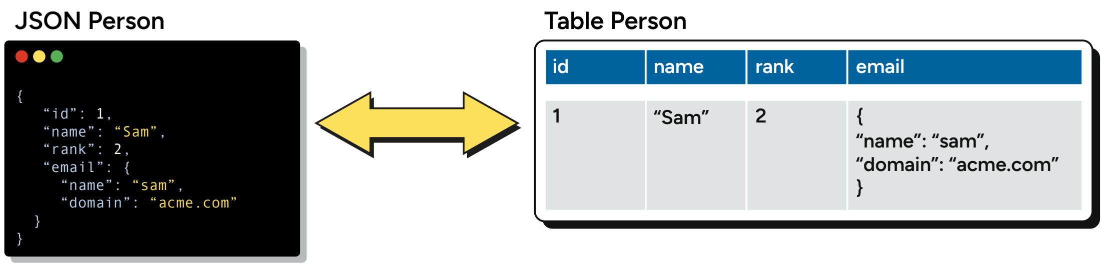
3	JSON DB API
4	New drivers for JSON DB API
5-8	Web Apps
9	Replication
10	Dynamic Data and Index Optimization
10	Dynamic Cache Resizing
11	Cache Priming at Server Startup
11-13	Security Enhancements
13	Recycle Bin
14	Optimistic Locking
15	Larger Default Index Page Size
15	Dynamic Plugin Management
15	Better Backups
16	Docker and Podman
16	Insert Only Tables
16	Read Only Servers
17	Improved Logging
17	Enhanced Record and Field Callbacks
18	Better Batch Operations
18	Fewer interruptions to service
18	File Enhancements
19	SQL Improvements
20	Embedded FairCom Engine (StandAlone)

DB v13

Key Feature: JSON DB API

FairCom DB v13 has a new API, JSON DB, that lets you quickly build web applications and services on your ISAM files using any programming language, such as Java, C#, Javascript, Python, etc. You don't need a driver because JSON DB uses JSON and HTTP, which all programming languages support.

All developers can quickly use the JSON DB API because it uses simple JSON commands and automatically maps between ISAM and JSON. For example, you can use JSON commands to create tables, query data, run SQL statements, etc. You can query an ISAM record as a JSON document and save a JSON document into an ISAM record.



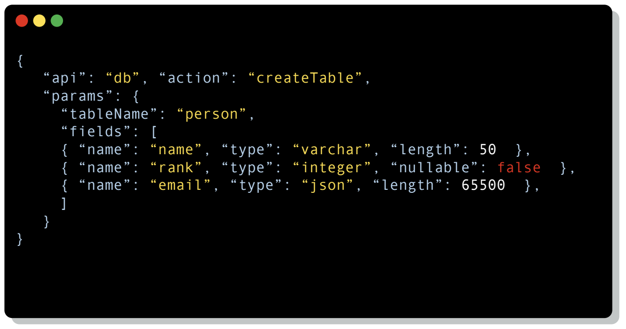
JSON Person

```
{
   “id”: 1,
   “name”: “Sam”,
   “rank”: 2,
   “email”: {
     “name”: “sam”,
     “domain”: “acme.com”
   }
}
```

Table Person

id	name	rank	email
1	“Sam”	2	{ “name”: “sam”, “domain”: “acme.com” }

JSON command to create a person table



```
{
   “api”: “db”, “action”: “createTable”,
   “params”: {
     “tableName”: “person”,
     “fields”: [
     { “name”: “name”, “type”: “varchar”, “length”: 50  },
     { “name”: “rank”, “type”: “integer”, “nullable”: false  },
     { “name”: “email”, “type”: “json”, “length”: 65500  },
     ]
   }
}
```

The JSON DB API manages all database operations:

- Create, alter, delete, list, and describe databases, tables, and indexes.
- Insert, update, and delete records using JSON.
- Run SQL queries and statements and return the results as JSON.
- Use specific JSON DB actions to query data with less code and faster results than SQL:
  - Look up records by key.
  - Tail the most recently inserted data.
  - Retrieve all records as quickly as possible.
  - Retrieve N records sorted ascending or descending.
  - Retrieve sample data by skipping through records.
  - Retrieve records matching a partial key.
  - Retrieve records between two keys.
  - Look up the closest matching record and retrieve its surrounding records.
  - Use a cursor to return paginated data efficiently.
  - Use a cursor to walk records forward or backward while skipping over data.

DB v13

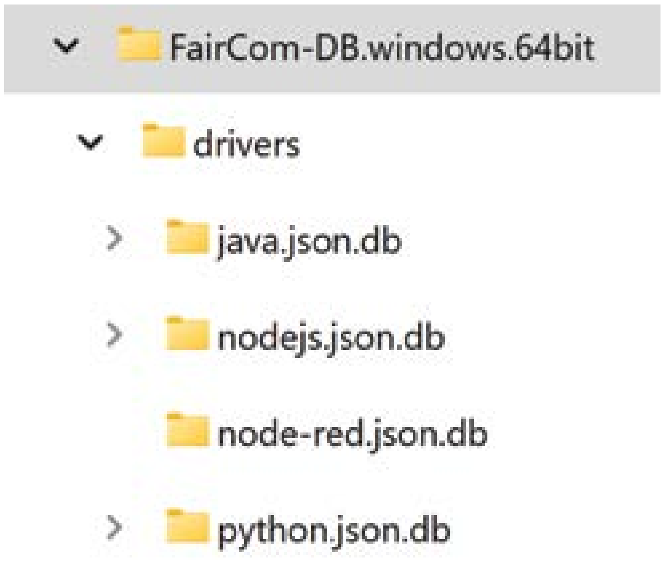
JSON DB API Tutorials

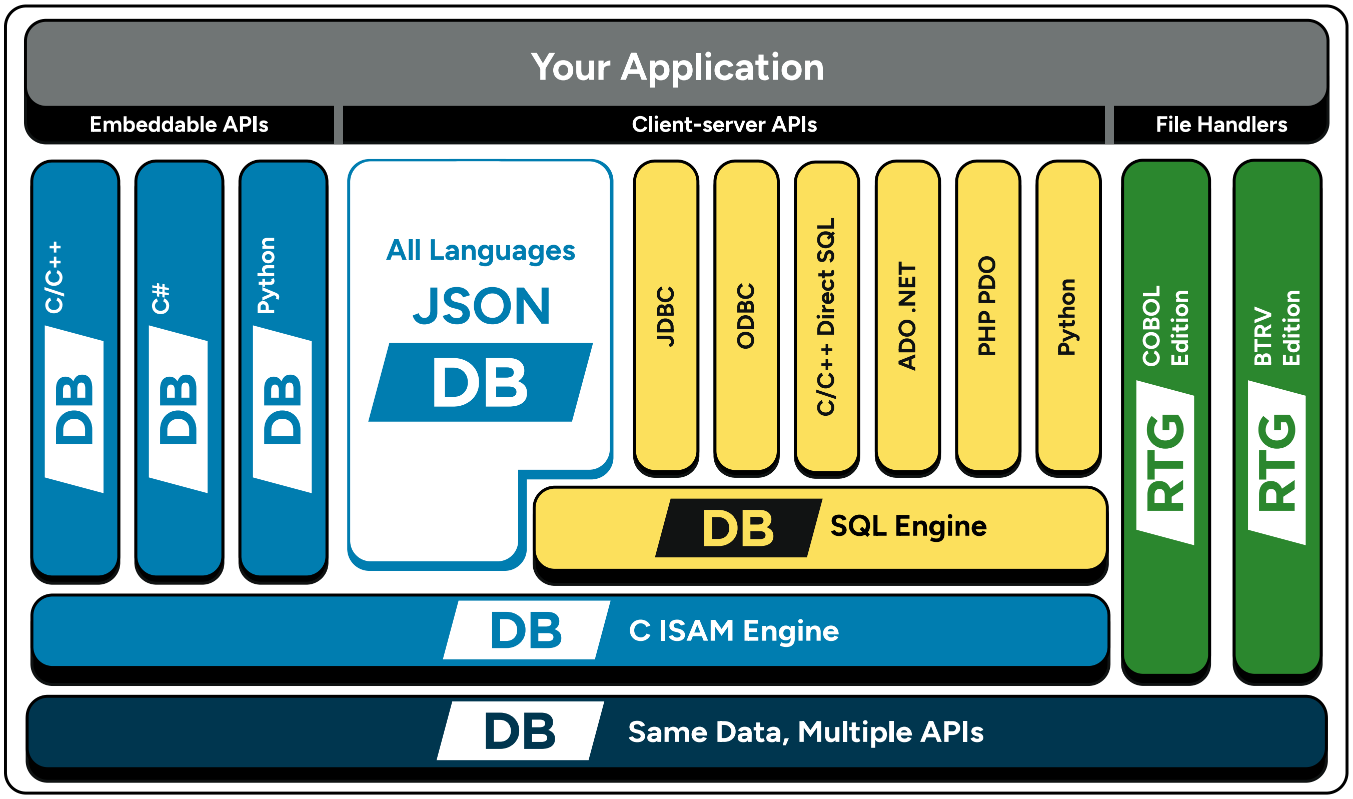
FairCom DB v13 includes tutorials for using the JSON DB API with JSON commands over HTTP.

To help you get started quickly, we've included JSON DB client libraries for the following languages:

- Java
- Node.js (JavaScript)
- Node-RED
- Python

You can find these tutorials in the <faircom>/ drivers folder:

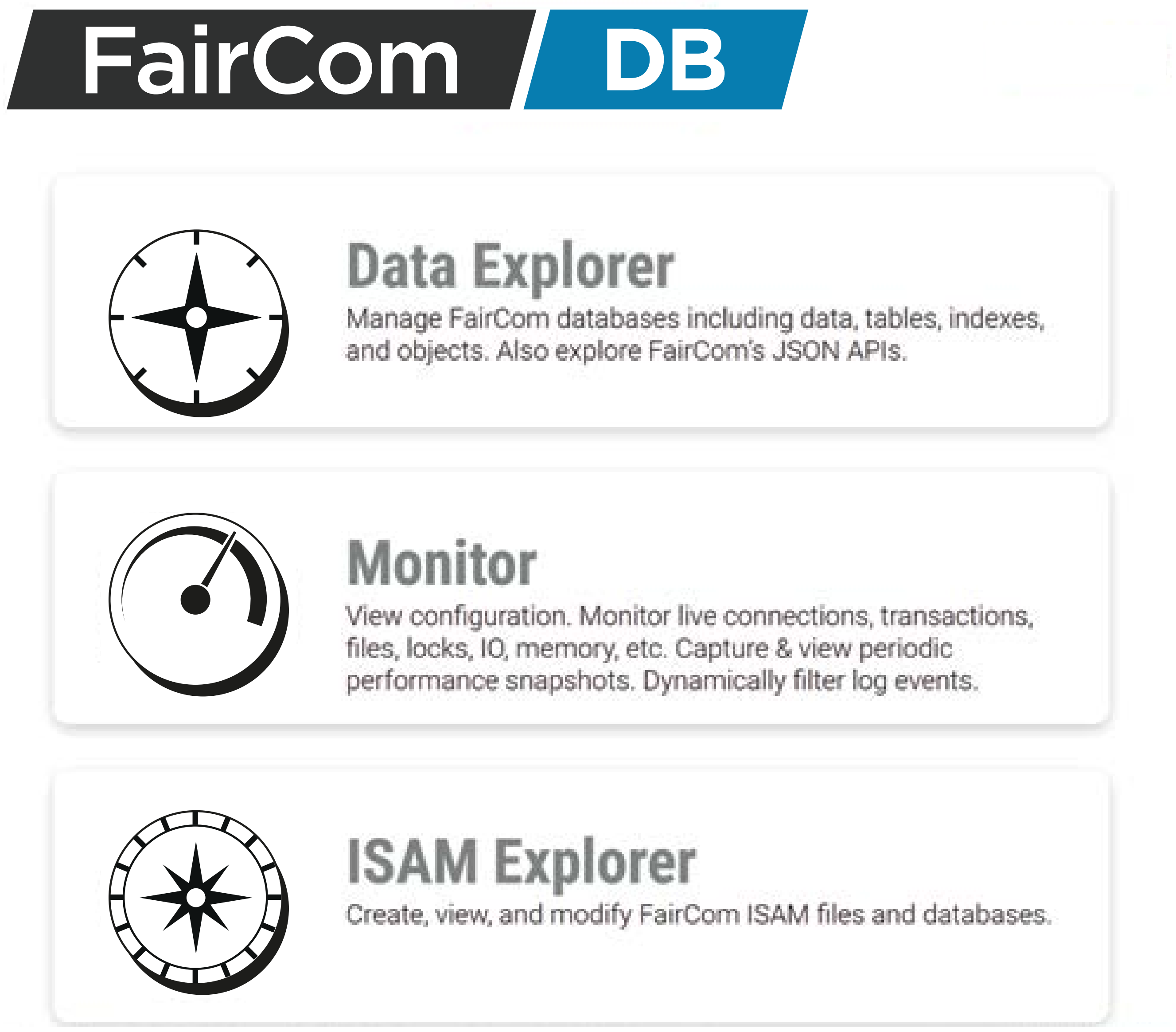




DB v13

Web Apps

FairCom provides three browser-based applications for DB: Data Explorer, Monitor, and ISAM Explorer.


FairCom DB

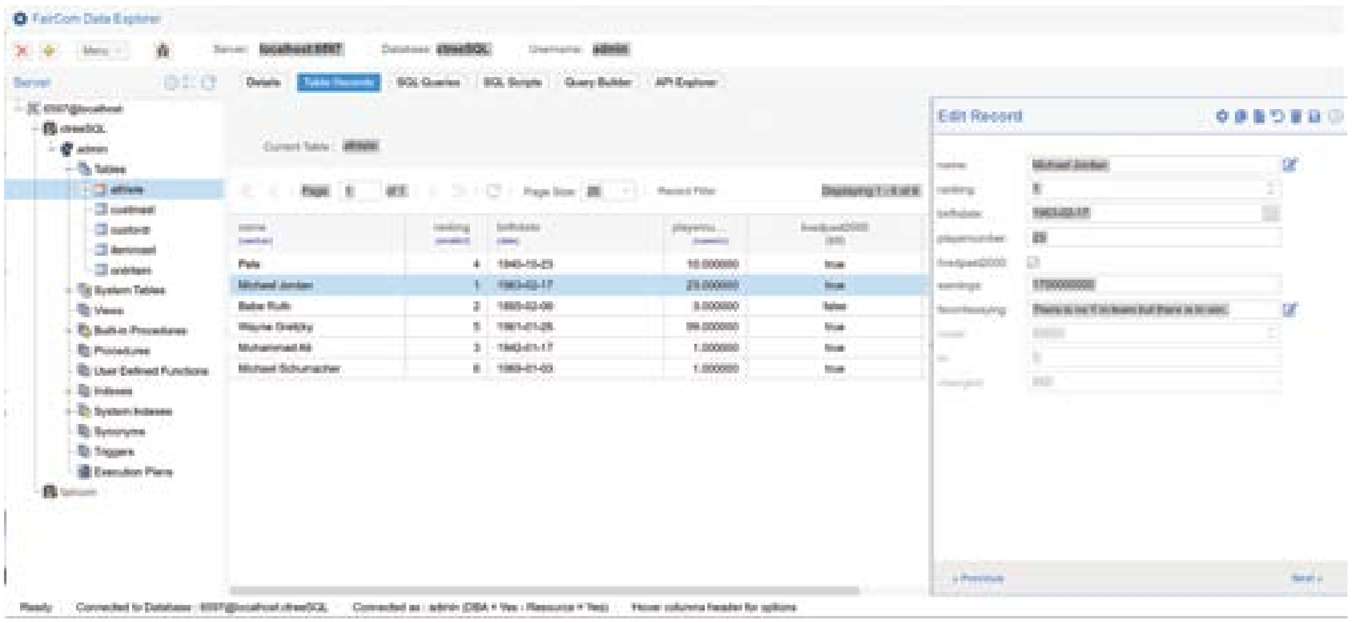
Data Explorer
Manage FairCom databases including data, tables, indexes, and objects. Also explore FairCom's JSON APIs.

Monitor
View configuration. Monitor live connections, transactions, files, locks, IO, memory, etc. Capture & view periodic performance snapshots. Dynamically filter log events.

ISAM Explorer
Create, view, and modify FairCom ISAM files and databases.

DB v13

Data Explorer manages your FairCom data and databases



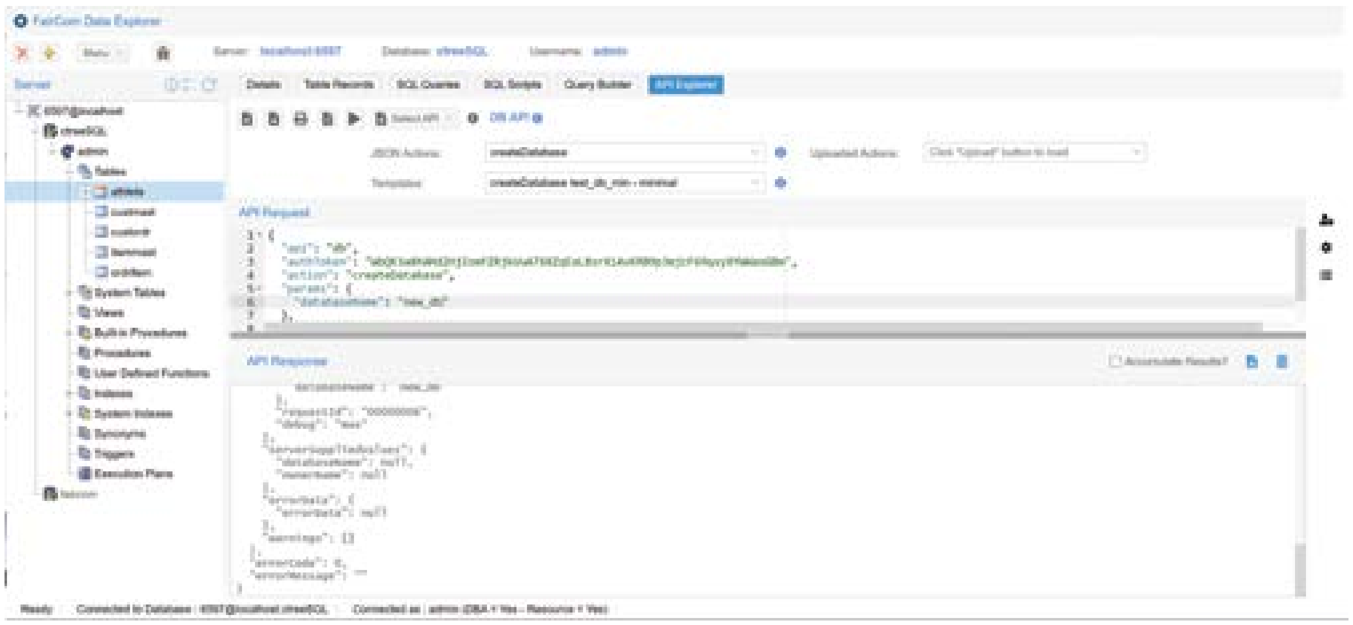
name	ranking	birthdate	playernu...	livedpast2000
Pele	4	1940-10-23	10.000000	true
Michael Jordan	1	1963-02-17	23.000000	true
Babe Ruth	2	1895-02-06	3.000000	false
Wayne Gretzky	5	1961-01-26	99.000000	true
Muhammad Ali	3	1942-01-17	1.000000	true
Michael Schumacher	6	1969-01-03	1.000000	true

Version 13 includes hundreds of improvements to Data Explorer. Creating, altering, and deleting tables, indexes, and SQL views is easier than ever. Query data using SQL and run any SQL statement.

Data Explorer has always been able to insert, update, and delete records, and now it includes a built-in binary editor to edit binary data. Use the HEX editor to change any byte of a binary value. View images embedded in binary fields. Save the binary content of any field to disk, edit it on disk, and upload the result right back into the field. Data Explorer's new Form View simplifies viewing and editing hundreds of fields in a single record.

DB v13

API Explorer manages databases with JSON



```
{
  "api": "db",
  "authToken": "[illegible]",
  "action": "createDatabase",
  "params": {
    "databaseName": "new_db"
  },
```

```
    "databaseName": "new_db"
  },
  "requestId": "00000008",
  "debug": "max"
 },
 "serverSuppliedValues": {
  "databaseName": null,
  "ownerName": null
 },
 "errorData": {
  "errorData": null
 },
 "warnings": []
},
"errorCode": 0,
"errorMessage": ""
}
```

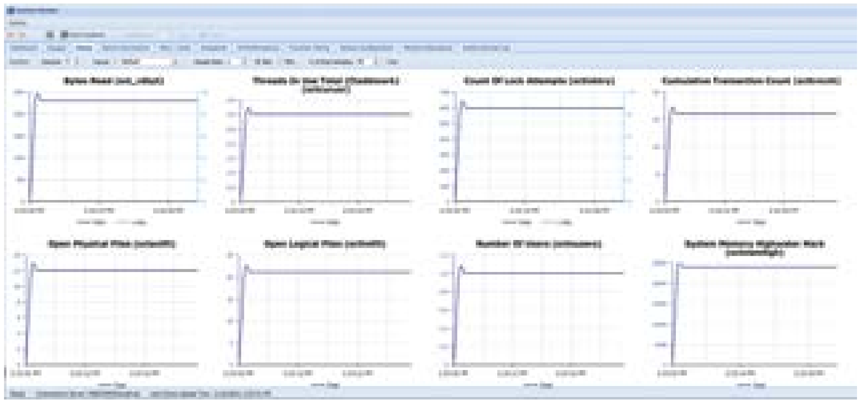
FairCom DB v13 adds a new API Explorer tab to Data Explorer. This integrated development environment for JSON APIs lets you run simple JSON commands to perform all database and administrative operations. It also provides integrated help and settings for each command.

You can copy the JSON commands you create with API Explorer and run them in Python scripts to automate database processes. Build entire applications by embedding JSON commands in your favorite programming language.

Use the Monitor Application in your Ops Center

The Ops Center provides hundreds of metrics, and you can create and save pages to monitor those that matter most. Use Monitor to display live dashboards on multiple monitors in your Ops Centers.

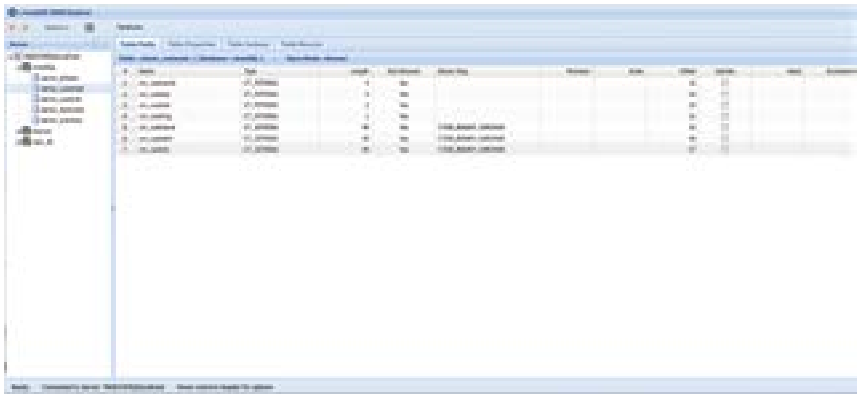
It can also help troubleshoot and solve issues like slow performance, long-running locks, users consuming too many resources, long-running queries, etc.



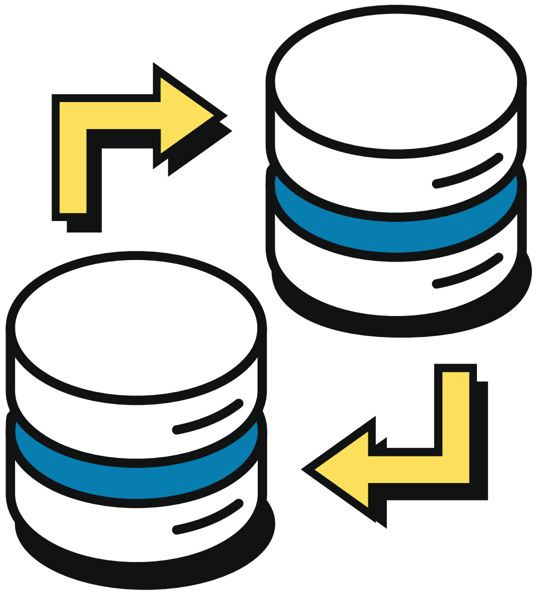
Use ISAM Explorer to see and manage your data files from the ISAM perspective

ISAM Explorer handles data files from the ISAM perspective. You can view and set the ISAM properties on a table or index.

In contrast, Data Explorer works with tables and indexes from the SQL and JSON DB perspective.



Replication



Failover and high availability are critical requirements for successful enterprise applications.

FairCom DB v13 supports Linux Pacemaker and Microsoft Server Clusters. You can combine replication with these high-availability solutions to create multiple read-only servers to scale your application and create disaster recovery servers.

You can create groups of files and manage their replications as a single unit. You can also use the new Replication Agent Manager to manage many replication agents to scale the replication process.

FairCom customers use this feature to increase scale and high availability in large-scale, multitenant solutions.

We encourage existing replication users to update to v13 and take advantage of many performance gains and stability improvements. Parallel replication greatly speeds the replication process.

The new replication command line utility (repadm) makes it surprisingly simple to start replicating data between FairCom servers. repadm makes it easy to script replication setup and management. The improved Replication Manager configures, starts, stops, monitors, and troubleshoots replication processes.

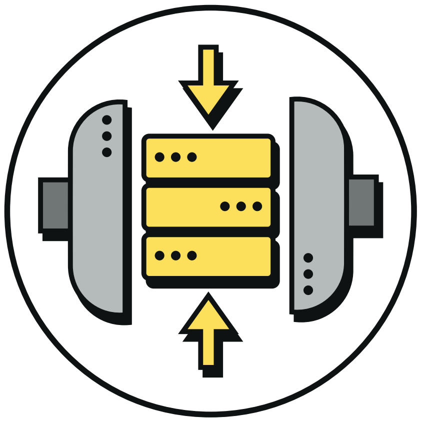
FairCom replication is versatile. You can replicate individual data files, superfiles, folders, and databases. You can use regular expressions to dynamically choose which files to replicate. When you replicate folders and databases, the replication engine detects when you add new data files to the folders and automatically replicates them.

FairCom replication supports many use cases.
• Replicate data synchronously to create highly available solutions.

• Replicate data asynchronously to create readonly servers and disaster recovery solutions.

• Replicate data bidirectionally and asynchronously to create horizontally scaled, eventually consistent solutions.

DB v13



Dynamic Data and Index Optimization

DB v13 lets your application compact and rebuild specific index and data files on the fly without affecting database operations—including reading and writing to those very same files.

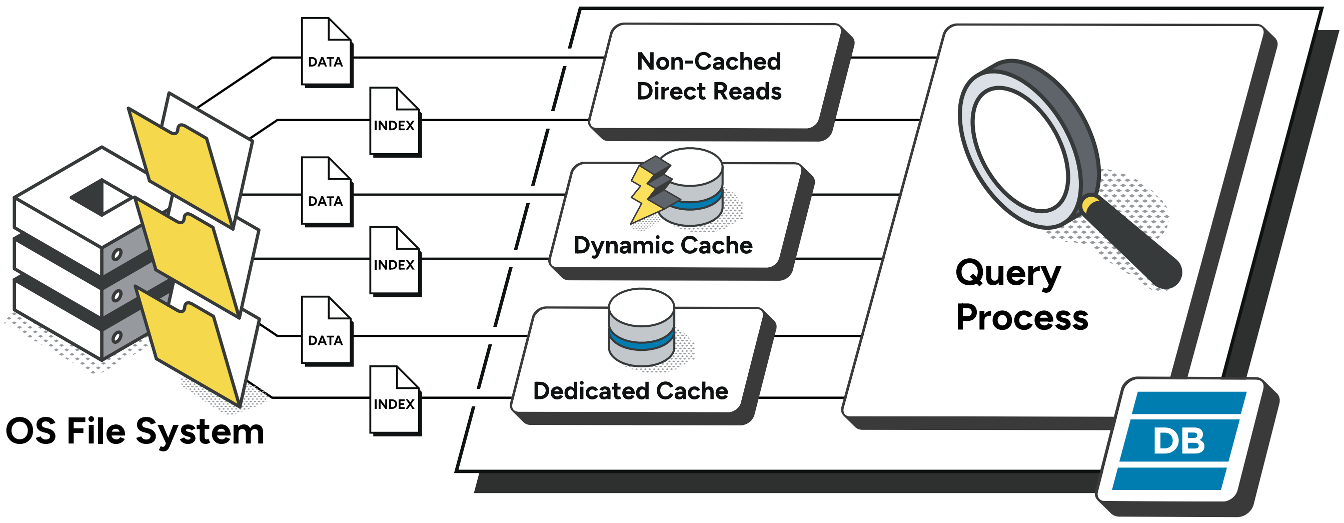
Cache Enhancements:

Dynamic Cache Resizing

Data caching is one of the most important database subsystems, and DB is no exception. Caching allows the server to quickly process data in RAM instead of retrieving it from disk.

Now, your application can resize the FairCom server's data and index caches on demand without restarting the server. This ability is important when the host does not have enough RAM to support the simultaneous needs of the server, applications, and OS caches.

The cache size can also dynamically increase as needed for additional database connections or before running an intensive data processing task. You can also have your app dynamically decrease the cache size to return RAM to the OS.



Cache Priming at Server Startup

FairCom servers have long-supported loading data into the data cache when an application opens a file. We call this "cache priming."

FairCom DB v13 now lets you prime the cache when the server starts by adding the subsystem cache command prime_cache_at_startup to ctsrvr.cfg.

This command configures the FairCom server to use a multi-threaded background process to load specific files and indexes into the cache during server startup. You can also adjust cache allocation per file and optionally configure record loading to forward or reverse index order.

Caching at server startup gives the FairCom server a high-speed, in-memory cache with predictable, low-latency response times.

Security Enhancements:

OpenSSL v3

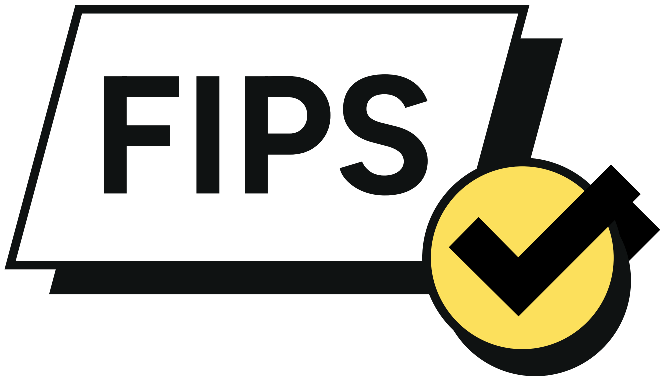
FairCom DB v13 uses OpenSSL 3.0 for all its secure communications, including TCP, HTTPS, WebSocket Secure (WSS), and MQTT Secure (MQTTS).

OpenSSL
Cryptography and SSL/TLS Toolkit

FairCom products are FIPS-ready

Federal Information Processing Standards (FIPS) are standards that protects sensitive or valuable data by verifying software compliance with its cryptographic requirements.

FairCom uses the OpenSSL toolkit to encrypt communications in transit and at rest. OpenSSL integrates with the FIPS-verification technology, and FairCom has completed all necessary integrations with OpenSSL to support the FIPS 140–2 standard.



DB v13

Client certificate authentication

Client applications can now authenticate with FairCom products using X.509 client certificates signed by your organization's private certificate authority. Client certificates and their private key replace passwords and are more secure because they are tamperproof, prevent brute-force password attacks, and expire. FairCom command line utilities now support client certificates in addition to usernames and passwords.

Client certificates are tamper proof because the organization cryptographically signs them. Client certificates prevent brute-force attacks because an attacker cannot guess the private key and cannot impersonate the certificate. Client certificates expire to limit the time window of an attack.

Private certificate management

You must use certificates to create secure connections to the FairCom server. Secure connections prevent malicious users from impersonating your servers and from listening in to communications between your servers.

FairCom provides Python scripts and tutorials to easily create and manage certificates. You can create your own private certificate authority, create server certificates, create client certificates, and renew expired certificates.

You aren't required to use Python scripts, and your security team can create a separate server certificate and private key file for each FairCom server. They're also able to create client certificates to authenticate your software with a FairCom server.

Secure communications by default

To ensure secure communications by default, a v13 FairCom server will only accept connections that use secure encryption algorithms.

You can still configure a FairCom server to support less secure or insecure communications as needed.

FairCom servers enforce HSTS by default

FairCom servers include an application server that accepts web connections. DB v13 servers support the HTTP Strict Transport Security (HSTS) protocol, which prevents insecure connections.

You can override this behavior to allow insecure connections like HTTP, WS, and MQTT.

Secure ECDHE connections

FairCom servers support SSL connections using ECDHE when Diffie-Hellman parameters are present in a server certificate file.

Elliptic-curve Diffie–Hellman Ephemeral (ECDHE) is a key exchange algorithm that allows two parties to establish a shared secret over an insecure communication channel. It is a variant of the Diffie–Hellman key exchange that uses elliptic-curve cryptography to provide more robust security with smaller key sizes.

Signed Windows and MacOS binaries

As a registered and notarized developer for Windows and Apple applications, all FairCom products sign their binaries for Windows and MacOS. These signatures ensure the FairCom product you download comes from FairCom and has not been tampered with.

LDAP

As a reminder, FairCom servers have long supported LDAP (Lightweight Directory Access Protocol), an external service that authenticates and authorizes a username and password combination. FairCom supports MS Active Directory, OpenLDAP, and OpenDJ.

Recycle Bin

Configurable recycle bin protects against accidental table deletions.

Mistakes happen! DBAs and operators accidentally delete tables; Applications have bugs that delete tables. Sometimes you can restore from a backup, but that data is typically outdated.

Protect your work by configuring FairCom DB v13 to delete a file to a recycle bin instead of from disk. The server moves deleted files to a designated archive folder where you can restore them. A configurable background task runs periodically to permanently remove previously deleted files according to your retention policy.

Optimistic Locking

The new Optimistic Locking feature in FairCom DB v13 allows your application to update records safely without the complexity and performance overhead of locking records. Optimistic locking increases the speed of application development, the number of concurrent users, and data processing.

When multiple users use the same database, developers must protect against multiple users simultaneously writing to and reading from the same record. The traditional approach requires an application to lock a record to prevent other users from writing or reading it. This approach is called pessimistic locking because it assumes an application must prevent other users from writing or reading a record it is processing.

Pessimistic record locks slow processing because applications must wait for other applications to release locks. For example, a user may cause an application to lock a record and go away for a long time, or an application may have a bug that locks a record and does not unlock it.

Optimistic locking solves these problems by not locking records. Instead, when an application reads a record, modifies it, and sends it to the server to be updated, the server checks to see if the record has changed between the time the application read the record and requested the update. If no other process changed the record, the server accepts the update. If another process changed the record, the server returns an error, and the application may re-read the record, reconcile the changes, and optionally resubmit the update.

Optimistic locking is preferable to pessimistic locking. However, it has been complex for databases and applications because they must implement a mechanism to detect a record change between when the application reads the record and requests the update.

FairCom DB v13 makes optimistic locking easy by enabling you to add a new optional field named changeid to tables that have enabled transaction processing. Each time a record is updated, the server puts the transaction ID of the update into the changeid field. When an application reads and updates a record, the record includes the changeid field. When the server receives an update request, it looks at the value of the changeid field in the update request and compares it to the current value of the changeid field in the record. If it is the same, the server allows the record to be updated; otherwise, the server returns error 1192 because another process changed the record.

By default, the new JSON DB API supports optimistic locking by automatically creating tables with the changeid field. When an update fails, the JSON DB API returns the original and updated record values to the application for reconciliation and resubmission to the server.

An application may not want optimistic locking. In that case, it can create tables without the changeid field or without transaction processing, or omit the changeid field from its queries and updates.

DB v13

Larger Default Index Page Size

DB v13 notifies you when you need to rebuild index files to increase performance

Most operating and storage systems are optimized to process data in larger chunks, which is why applications benefit when FairCom DB processes data with a page size of 32K or 64K. FairCom DB v13 defaults to a page size of 32K, but you can change this in ctsrvr.cfg.

In v13, FairCom DB automatically adjusts the page size of its internal files (dictionaries, *.FCS files, etc.) to match the page size in ctsrvr.cfg. This increases data processing speed. When the server opens a file with a smaller page size, it adds a warning log entry to CTSTATUS.FCS. Keep an eye out for these warnings, and convert application indexes to use the larger page size as needed.

Dynamic Plugin Management

Start, stop, and change plugins while the server is running

You don't need to stop and restart your servers when managing plugins! FairCom DB v13 has a command-line utility and API calls to dynamically start, stop, and change the settings for individual plugins while the server is running.

Better Backups

Back up your server without waiting for current transactions to complete

FairCom DB v13 no longer requires backups to wait until all outstanding transactions are complete. Add !ALLOW_TRAN to the dynamic dump script to start a backup without waiting for transactions to finish. This setting increases the backup size proportional to the active transaction size at the backup time. It also requires additional processing when restoring the backup.

Restore files to a specific path

Restore files to a different path than the original backup path, which is great when restoring a backup for a new database. Use the !REDIRECT option in a dynamic dump restore script to redirect all directory names to a specified directory name. The new directory name can be a full or a relative path.

Improved performance of backup status messages

FairCom DB v13 delivers backup status messages efficiently and no longer slows the backup process when the client is on a different computer than the server.

Docker and Podman

Run FairCom DB in Docker and Podman for deployment to hosting services, such as Kubernetes

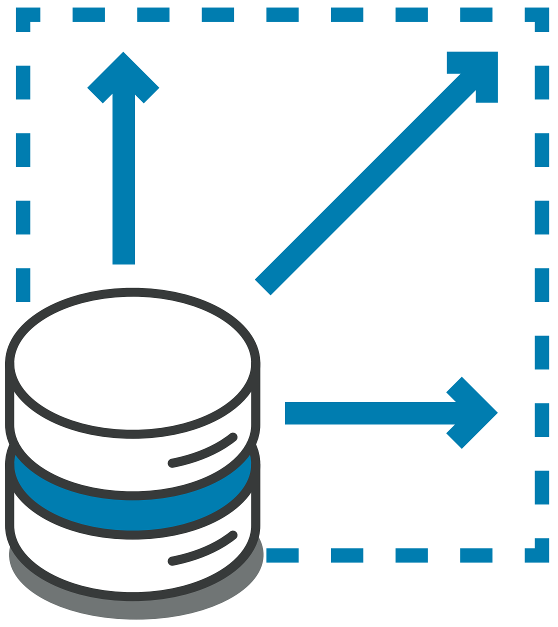
Creating a Docker or Podman image of FairCom DB is a simple process. To make it even easier, FairCom DB v13 now provides a tutorial and utility to help automate the process.

Insert-Only Tables

Create tamper-proof tables

When you need a tamper-proof historical record, such as tracking user actions or collecting time series data from a device, you can use the ctInsertOnly file mode to create an insert-only table that prevents records from being deleted or updated—even if you grant a user update or delete privileges on the table. Users with insert and read privileges can still insert and read records from an insert-only table.

Read-Only Servers



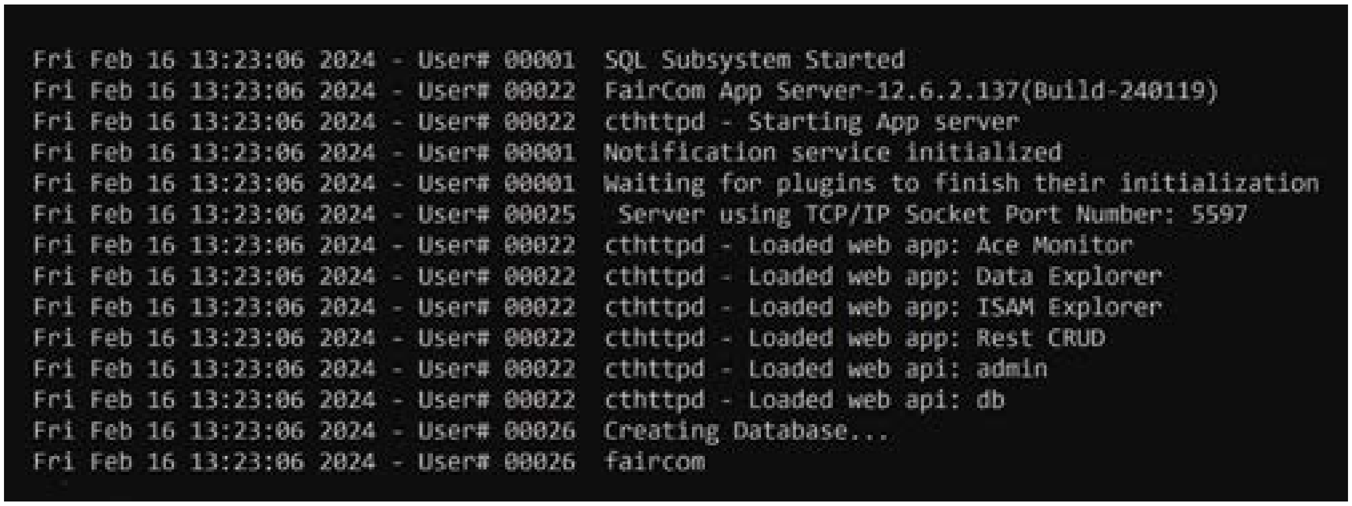
For high availability, load balancing, and scaling solutions

Create multiple read-only servers and use data replication to keep them in sync with a source server. This lets you implement a load balancing solution or high availability while also increasing scalability. You can also populate a server with data and make it read-only to keep end users from changing its data.

A read-only server blocks normal users and applications from inserting, updating, and deleting records on the server. It does not stop data replication from synchronizing changes from a source server to a read-only server. The server logs a line to the CTSTATUS.FCS file when Read-Only Mode is turned on or off.

DB v13

Improved Logging



```
Fri Feb 16 13:23:06 2024 - User# 00001  SQL Subsystem Started
Fri Feb 16 13:23:06 2024 - User# 00022  FairCom App Server-12.6.2.137(Build-240119)
Fri Feb 16 13:23:06 2024 - User# 00022  cthttpd - Starting App server
Fri Feb 16 13:23:06 2024 - User# 00001  Notification service initialized
Fri Feb 16 13:23:06 2024 - User# 00001  Waiting for plugins to finish their initialization
Fri Feb 16 13:23:06 2024 - User# 00025   Server using TCP/IP Socket Port Number: 5597
Fri Feb 16 13:23:06 2024 - User# 00022  cthttpd - Loaded web app: Ace Monitor
Fri Feb 16 13:23:06 2024 - User# 00022  cthttpd - Loaded web app: Data Explorer
Fri Feb 16 13:23:06 2024 - User# 00022  cthttpd - Loaded web app: ISAM Explorer
Fri Feb 16 13:23:06 2024 - User# 00022  cthttpd - Loaded web app: Rest CRUD
Fri Feb 16 13:23:06 2024 - User# 00022  cthttpd - Loaded web api: admin
Fri Feb 16 13:23:06 2024 - User# 00022  cthttpd - Loaded web api: db
Fri Feb 16 13:23:06 2024 - User# 00026  Creating Database...
Fri Feb 16 13:23:06 2024 - User# 00026  faircom
```

Each log entry is now a single line

The CTSTATUS.FCS status file defaults to a single-line format, making reading and parsing much easier.

System logs grow larger and purge faster

In v13, system log files—SYSLOGDT.FCS and SYSLOGIX.FCS (see SYSLOG)— can grow up to 100 million terabytes and be completely purged in seconds.

Enhanced Record and Field Callbacks

FairCom DB v13 allows you to write callbacks in C to preprocess records. You can modify data before the server stores it in the database and before the server returns it.

You can also create and register a custom function to convert field values when the Hot Alter Table process updates a record.

DB also lets you register table callbacks that work at the record and field levels.

It uses a fallback pattern to run callbacks, making them more manageable. For example, when you don't register a callback library at the table level, the server uses the last used callback library. If you never specified a callback library, the server returns an error.

DB v13

Better Batch Operations

Reuse buffers to eliminate the cost of allocating and deallocating RAM

The new batch mode CTBATCH_KEEPBUFFER (or for CTDB Programmers) reuses a buffer for subsequent batch calls. Use this when you know subsequent calls to a batch process will fit into the size of the original buffer.

- In v13, batch operations run faster because they reduce server roundtrips, filter field expressions faster, and have the buffer functions return the number of skipped filtered records.

Fewer Service Interruptions

Friendly file blocking

Applications with admin permissions can block file access to perform maintenance routines requiring access exclusivity. Before v13, blocking a file immediately terminated all transactions running on it. Now, the server can optionally wait for currently running transactions on the file to finish within a timeout window before blocking new transactions. See ctFIIBLK() and ctFILBLKX().

Modify an index filter while the data file is in use

DB v13 lets applications add a conditional filter to an index while the data file is open in shared mode. Before v13, adding a conditional index required the file to be open in exclusive mode.

File Enhancements

Create a small data file and let the server increase the file size upon inserting the first record

The server can quickly insert large numbers of records into a data file when it increases the size of the data file in large chunks, called extents. You can configure the extent size (to set for Index File) of each data file (IFIL.dxtdsiz). Often, though, you have data files that you want to be small until you insert the first record. DB v13 meets both needs by allowing you to set a small initial extent and a large growth extent.

Use file names that are 4096 bytes long

An operating system may limit a file name and path to less than 4096 bytes.

Open 1 million files at once

By default, FairCom DB v13 can open 1 million files simultaneously and can be configured to reach even greater numbers.

DB v13

SQL Improvements

Faster SQL performance

- Large string fields—up to 65,500 bytes—allocate memory more efficiently for faster performance.
- Converting the NUMBER type to BIGINT is now 100x faster.

More capable SQL

- SQL and all other FairCom APIs can specify when the server sets the value of an autotimestamp field. The server can set the timestamp when you insert a record, update a record, or insert and update a record.
- Use SYSLOG_EXCLUDE_SQL_USER to configure the server to turn off SQL logging for specific users, such as system processes that fill the logs with unhelpful events.
- Stored procedures can access error codes and messages using the DhSqlException.getSqlState() and DhSqlException.getMessageText() functions.

Improvements to direct SQL API for C and C++ programmers.

- FairCom has added new cursor functions to the Direct SQL API. Cursor functions efficiently retrieve records from a SQL query by tracking the cursor's current position. Cursors can efficiently paginate data, tail the end of a file, retrieve records forward and backward, get the top N records, skip records forward or backward, and repeatedly jump to the first and last record.
- We've also added new functions to the Direct SQL API for sending in parameters to stored procedures and setting the value of output parameters.

DB v13

Embed the FairCom Engine with DB Standalone

FairCom continues to offer the most capable embedded database on the market.

- FairCom's DB Standalone service allows FairCom's Standalone Multi-User engine to be embedded directly into the application through linking or compiling. It also enables thousands of third-party SQL applications to use ODBC and JDBC to run SQL queries.
- Keep costs low even while scaling office applications to scores of simultaneous users on Windows, Linux, Unix, and MacOS.
- DB Standalone service combines a SQL engine with FairCom's most popular embedded model, Standalone Multi-User. This non-client-server model supports multiple users reading and writing simultaneously to the same set of files—including SQL users connecting through ODBC and JDBC—by forcing each write and read directly to disk.
- DB v13 adds partitioned files to the Standalone Multi-User model and works in the single-threaded and multi-threaded versions. Partitioned files automatically retain data for specific periods, such as keeping data for 30 days and purging data older than 30 days.
- DB v13 also adds rebuild and compact utilities for data encryption.

Promote a Multi-User ctREADFIL and ctSHARED open to a shared read-only open: When using Unix systems with the Standalone Multi-User operational model, if a process has a file open in ctSHARED mode, another process attempting to open the file in ctREADFIL | ctSHARED mode fails with error 920 (and vice versa).

This is by design, because in Standalone Multi-User mode the ctREADFIL | ctSHARED mode is converted to ctREADFIL by removing ctSHARED from the file mode, and ctREADFIL opens are not compatible with the ctSHARED filemode. This is because ctREADFIL caches index node reads, so the Standalone Multi-User library does not permit other opens with write access when the file is open with ctREADFIL.

We introduced a compile-time option: #define ctBEHAV_FPG_SHARED_READFIL causes a ctREADFIL | ctSHARED open to be treated as a shared open with read-only access. Note: This means that since other processes are allowed to open the file for write access, this type of shared read-only open does not enable caching.

FairCom

DB

v13

DB Made Simple

FairCom