test link

Release Notes

c-treeACE V10.3 Release Notes

Audience

Developers

Subject

FairCom's high-performance NAV and SQL database technology.

© Copyright 2025, FairCom Corporation. All rights reserved. For full information, see the FairCom Copyright Notice (page lxii).

FairCom®

Contents

All Rights Reserved

All Rights Reserved

All Rights Reserved

All Rights Reserved

All Rights Reserved

All Rights Reserved

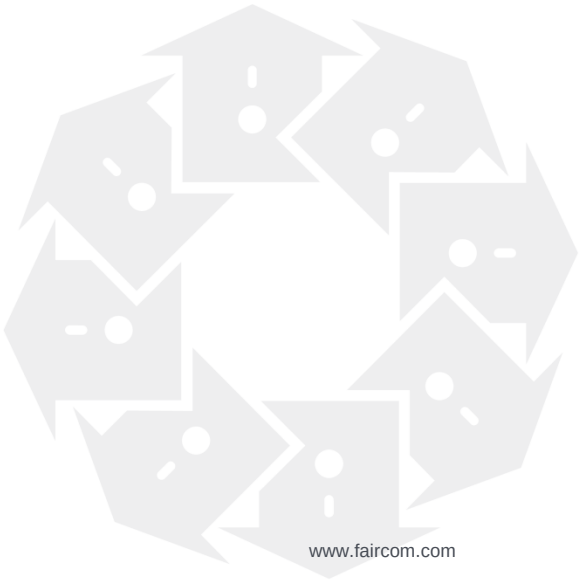
1. Introduction

We are pleased to deliver another c-treeACE release to our valued maintenance customers. Many improvements and corrections have been implemented to make our latest release the best ever.

This document lists the corrections we have implemented in this release. These changes are grouped by product area.

You may have received some of these changes in a prior “interim” delivery. The best way to ensure that you get the latest set of changes is to update to this, our latest release.

V10.3 is packed with new features—from simple enhancements to major breakthroughs. The new and updated features are listed in the V10.3 Update Guide (https://docs.faircom.com/doc/v103ace/).



All Rights Reserved

2. Important Changes

This section lists important changes included in the latest version of c-treeACE.

2.1 Automatic Recovery - Bus error during automatic recovery

In very rare situations during recovery, while determining if a missing file was actually required for recovery, a variable might not be properly aligned, causing the server to crash with a **SIGBUS** error on selected architectures. The logic has been corrected to eliminate this issue.

2.2 Automatic Recovery - Superfile TRANDEP rename

This modification corrects an obscure situation involving an L64 c-treeACE Server “catend” operation that occurred when a SUPERFILE with a flag indicating the header record was not valid was encountered during automatic recovery of a transaction dependant rename operation.

2.3 Concurrent calls to ctdbAllocSession() crashing on Windows

A crash was occasionally seen on Windows systems in a small number of situations when multiple threads were making concurrent calls to **ctdbAllocSession()**. The logic has been changed to correct this.

2.4 Dynamic Dump / Segmented File deadlock

A Dynamic Dump of a segmented file was causing a deadlock situation because the dump was trying to create a segment on a read operation. This modification detects when a dump thread is in a read operation and skips the create-segment operation.

All Rights Reserved

3. General Fixes

3.1 Adding identity field after adding records causes record update to fail

When an identity field was added to the variable-length part of a c-tree data file and then a record was updated before any records were added, the record update failed with error **FMEM_ERR**. (In earlier versions, the record update caused a memory overwrite that caused c-treeACE Server to terminate with an unhandled exception.) The logic has been modified to correct this.

3.2 NCR Unix - Ported to new platform and fixed system error message from random number generator

A customer requested our COBOL database support for NCR Unix. Due to the highly portable nature of our design, this support was implemented with a few simple changes. If you have need for support for any older Unix Systems, feel free to check with your local FairCom office about possible availability. We take pride in offering c-tree support on a wide array of systems.

3.3 Check for invalid FORCE_LOGIDX and COALESCE_TRNLOG configuration options

The FORCE_LOGIDX and COALESCE_TRNLOG options did not check for an invalid value. If an invalid value was specified, the default option of NO was used. This caused unexpected behavior for one customer who used FORCE_LOGIDX YES when they meant to use FORCE_LOGIDX ON. As a result, this was treated as FORCE_LOGIDX NO, and LOGIDX was not forced on for the customer's TRNLOG index files, causing automatic recovery to take a long time for large index files.

Now c-treeACE Server checks if an invalid value is specified for these options and, if so, it fails its start and logs a message such as the following to *CTSTATUS.FCS*:

```
Configuration error: ctsrvr.cfg, line 1: Invalid value specified for FORCE_LOGIDX option. Valid values are ON, OFF, or NO.
Configuration error: ctsrvr.cfg, line 3: Invalid value specified for COALESCE_TRNLOG option. Valid values are ON or OFF.
```

All Rights Reserved

3.4 c-tree client library now avoids using socket after connection to server has been lost

On Unix systems, when a client lost the connection to c-treeACE Server, if it made another call to the server, the socket call raised an **SIGPIPE** exception. If the calling program did not handle this signal, the client process terminated. This was a problem for a program that did not want to handle the **SIGPIPE** exception but wanted to call **CLISAM()** and then reconnect to c-treeACE Server.

A state bit is now set in the client library when the connection has been lost. On subsequent calls, the socket is not used. If a function such as **CLISAM()** is called to close the connection, the client library cleans up the connection so that the client can reconnect.

3.5 c-treeACE Server fails to start on Unix with "Missing License File" error

On a Unix system, when c-treeACE Server started in a directory that contained a file that was a symbolic link to a non-existent file and that filename preceded the name of any license file in alphabetical order, c-treeACE Server erred out on that file and returned a **Missing License File** error. The logic has been changed to correct this.

3.6 c-treeACE Server on Unix system looks for license file in LOCAL_DIRECTORY directory

On a Unix system, if c-treeACE Server logged a message to *CTSTATUS.FCS* before checking the license, it looked for the license file in LOCAL_DIRECTORY. This caused c-treeACE Server to fail to start with an error indicating that it did not find the license file, although the license file existed. The logic has been modified to correct this.

3.7 c-treeACE Server - Unhandled exception freeing memory during checkpoint with replication

Under very rare circumstances, the c-treeACE Server process could terminate with an unhandled exception when freeing memory during a checkpoint operation if replication was enabled on a file involved in the active checkpoint operation. The logic has been modified to minimize this problem; a mutex has been introduced to ensure that a file's replication status cannot change during any critical operations within the checkpoint.

All Rights Reserved

3.8 c-treeACE Server - Unhandled exception when REPL_MAPPINGS configuration option is used

c-treeACE Server was seen to terminate with an unhandled exception periodically when the `REPL_MAPPINGS` configuration option was used in conjunction with a **PUTHDR()** call. The logic has been modified to handle this situation.

3.9 c-treeACE Server - Unhandled exception when STPUSR or CLISAM is called before ending impersonation

When an ISAM connection impersonated a SQL connection, if the ISAM connection called **STPUSR()** or **CLISAM()** before ending the impersonation, c-treeACE Server terminated with an unhandled exception. The logic has been changed so that it undoes the impersonation completely before executing **STPUSR()** or **CLISAM()**, which eliminates the problem.

3.10 c-treeACE Server - Unhandled exception in RBLIFILz when rebuilding file with more than 32 indexes

When a file with more than 32 indexes was rebuilt, c-treeACE Server terminated with an unhandled exception in **RBLIFILz()**. The logic has been changed to correct this.

3.11 c-treeACE Bound Server - Unhandled exception when bound thread calls rebuild or compact with a callback function set

FairCom provides a powerful feature, the c-treeACE Bound Server model, which allows you to load the c-treeACE Server DLL in your own application space rather than calling c-treeACE functions through remote procedure calls in a separate process. When the c-treeACE Bound Server model sets a rebuild/compact callback function, an unhandled exception could occur during the rebuild or compact operation. The logic has been modified to correct this situation.

3.12 Compact utility fixed

During testing, we identified an issue in which the *-redosrl* option of **ctcmpcif** was ignored. This situation has been rectified by modifying the utility and internal c-tree logic. If the utility and the internal c-tree logic are not aligned (e.g. new client utility with old server) the utility fails with error **TFLN_ERR** (943).

All Rights Reserved

3.13 ctQTblockLogFiles - Avoid possibility of log buffer write when in force

A **ctcatend** condition was caused by trying to write to closed log files. When ctQTblockLogFiles is part of a ctQUIET request, the log files are actually closed at the system level. Even though ctQTblockLogFiles requires that ctQTblockAPI is in force and no transactions can be started, it was still possible for the quiet thread to cause log I/O to be attempted. The logic has been modified to correct this.

3.14 ctredirect of aidxnam names

An **IAIX_ERR** (608) was seen opening a file after running the **ctredirect** utility on the file. The redirect logic was changing aidxnam values from *FILENAME.idx* to *FILENAME.dat*. The logic has been modified to correct this issue.

3.15 ctsqlimp unlink not removing virtual tables

The **ctsqlimp** *unlink* logic (removal of a table linked into SQL) was not retrieving the list of virtual tables and, therefore, they were not listed for removal.

The **ctsqlimp** logic has been modified so that the list of virtual tables (MRT in particular) is retrieved from the dictionary itself during an unlink operation.

The *unlink* operation is not executed if the table cannot be opened in exclusive mode because someone else has it open.

3.16 cttcre leaks memory and sockets when it fails

Memory and SQL socket handles may be leaked when the maximum number of users is exceeded (**MUSR_ERR**). Prior to this fix, no error was returned to the client before the socket was closed if this situation occurred, so the client would receive a generic network error. The logic has been updated to properly return the **MUSR_ERR** error.

3.17 Dynamic Dump to detect system clock change

When a daylight saving time change (or other system clock change) occurred, a scheduled dynamic dump ran at the wrong time from that point forward. The logic has been updated to correct this: The dump thread now sleeps in one minute increments (or less than one minute when the scheduled time is less than one minute from the current time). When it awakes, the dump thread checks the current time and determines whether or not it needs to sleep again.

All Rights Reserved

3.18 Dynamic partition member bug fixes

The following errors have been corrected in the dynamic partition member support:

1. With `SQL_OPTION NO_CARDINALITY` in *ctsrvr.cfg*, an obscure set of conditions involving dynamic partition members could cause the server to crash.
2. c-treeACE Server did not detect a difference in index definitions between a dynamic partition host and a member.
3. After creating a unique index on the dynamic partition host using SQL, a query on the host and its members did not detect that the members did not have that index, and the query returned no rows.

3.19 Encrypt/Decrypt inconsistent for batch insert on AES-encrypted files

After using a batch insert to add records to a file that uses AES encryption, in rare cases reading the records failed with **error 160** because the records were not decrypted properly.

A second way to encounter this issue was by changing c-tree's page size after writing the data. If a different cache page size was used when reading AES-encrypted data than when the data was written, the data file contents would not decrypt properly.

The logic has been corrected and a new version of the AES cipher has been introduced to preserve the ability to read files that were created using the AES cipher prior to this revision.

When using Advanced Encryption, *FAIRCOM.FCS* is encrypted using AES. Older versions of this file can be read by the updated logic.

During this testing, the following improvements were made to the **ctscmp** utility:

1. **ctscmp** can now compact superfiles that use Advanced Encryption.
2. When **ctscmp** is compiled with support for transaction control, it can now compact transaction-dependent superfiles.

3.20 Error 127/128 using FSHAREMM on AIX

Intermittent **127/128** errors were seen when using shared memory on AIX. The logic has been modified to correct this situation.

3.21 Error 971 (SQL_MUSR_ERR) returned on an ISAM connection attempt

Error 971 (**SQL_MUSR_ERR**) was being returned on an ISAM connection attempt. This error was returned after the SQL connections timed-out if SQL clients were still connected.

All Rights Reserved

The logic that checks connection limits and expirations was checking the TOTAL number of ISAM and SQL connections without considering the connection type (ISAM or SQL). When either SQL or ISAM connections expired, the maximum number of allowed connections was set to 0, so the next attempt to connect would result in an error.

An **ISAM_MUSR_ERR** error could be returned when attempting a SQL connection with an expired ISAM connection.

The logic has been modified to correct this situation.

3.22 Failed file compact deletes original and leaves compacted file with temporary filename

When a file compact operation failed after deleting the original data file but before renaming the newly compacted data file, the new data file was left with its temporary filename. In this situation the following message was seen in *CTSTATUS.FCS*:

```
 - User# 00017 Compression: original deleted.
```

But the following message, which normally follows that first message, was not seen in *CTSTATUS.FCS*:

```
 - User# 00017 Compression: temporary renamed.
```

The logic has been change to eliminate this scenario (the original is not deleted until the temporary file has been renamed). With these changes, a successful file compact now logs the following messages to *CTSTATUS.FCS*:

```
 - User# 00017 Compression: original renamed.
 - User# 00017 Compression: temporary renamed.
```

When a file compact fails and an attempt to rename the original data file to its original name fails, c-tree logs a message to *CTSTATUS.FCS*:

```
Compression: could not restore original data file name: <error_code>
<temporary_filename>
<original_filename>
```

3.23 Failed GETRES() writes to output buffer in client/server mode

In client/server mode, a **GETRES()** call that failed (because no matching resource was found) wrote random bytes to the output buffer (*bufptr* parameter). The logic has been modified to correct this issue.

3.24 FAIRCOM.FCS upgrade problems with ADMIN_KEY

c-treeACE V10 Server startup has been seen to fail with the following *CTSTATUS.FCS* message:

```
Mon Aug 05 09:56:11 2013
 - User# 00001 FAIRCOM.FCS does not match settings file: check with system administrator.
```

This occurred when *all* of the following were true:

- ADMIN_KEY was specified in *ctsrvr.set*
- *FAIRCOM.FCS* was created by a prior version of c-tree
- ADMIN_KEY was over 9 bytes

In V10, the ADMIN_KEY value stored in *FAIRCOM.FCS* was increased from 9 bytes to 64 bytes. The logic has been modified to properly update the ADMIN_KEY to the new format.

3.25 File Create Error 753 (SPCL_ERR) when using ctAUTOSAVE and ctDEFERBEG

If **CREIFIL()** was used to create a transaction-controlled file following a call to **TRANBEG()** with the ctAUTOSAVE and ctDEFERBEG modes with no other transaction activity in between, **CREIFIL()** was failing with error 753 (**SPCL_ERR**, cannot restore or clear past special savepoint).

Note: The error happens only if using ctAUTOSAVE (with ctDEFERBEG) due to behavior involving ctAUTOSAVE:

The logic has been updated to correctly handle this situation.

3.26 Improved ctQUIET / Dynamic Dump interaction

Testing revealed that logic to prevent overlapping dumps when ctQUIET and dynamic dump threads interact was not employed as effectively as possible. The logic has been changed to avoid possible system hangs and improve the interaction.

3.27 Invalid DAR value crashes server

After an ALTER TABLE operation, UPDATE of a record with an identity in the variable portion of the record and a unique index on the identity column crashed the server. The logic has been changed to correct this.

3.28 ISAM client on high/low byte-order system does not attempt a shared memory connection when localhost host name is specified

On a high/low byte-order system, an ISAM client that was built with both TCP/IP and shared memory support did not attempt a shared memory connection when the host name was specified as `localhost`. If no host name was specified, or the system's host name (registered in the DNS) was specified, the client did attempt a shared memory connection.

Examples:

When no host name was specified, **ctadmn** attempted a shared memory connection:

```
ctadmn ADMIN ADMIN """" FAIRCOMS
```

When localhost was specified, **ctadmn** did not attempt a shared memory connection:

```
ctadmn ADMIN ADMIN """" FAIRCOMS@localhost
```

The logic has been modified to correct this.

3.29 License error if license file belongs to user/group not on the machine

A server license error (missing license file) was experienced on startup when the license file was in place, and readable by the "world," but the file owner and/or group did not exist at the OS level. The function that searched for the license file on UNIX systems returned an error if the user and/or group were not valid. This resulted in the license subsystem thinking that the license file was not present. The logic has been changed to correct this.

3.30 Linux FSHAREMM connection fails with packed alignment

When a Linux shared-memory client was compiled with 1-byte alignment, the connection failed with **LSEC_ERR** or another spurious error.

A Unix shared-memory structure SHRMEM_HDR differs in size between client and server depending on the compiler alignment, causing the data payload to be missing the first byte on the server. Padding bytes were added to properly align the structure in this condition.

3.31 ctcv67 final messages upon exit

When exiting, the final messages redirected from **ctcv67** output did not always appear in the output file. The logic has been changed to correct this.

3.32 Low-level ctOPENCRPT open of replicated data file fails with error 775

Attempting to open a replicated data file using a low-level file open with the ctOPENCRPT file mode failed with error 775 (**UNQK_ERR**, no UNQKEY support for REPLICATION) if the file was open in shared mode by the KEEPOPEN_LIST option. The logic has been updated to account for the KEEPOPEN_LIST status.

All Rights Reserved

3.33 PUTDODA in standalone mode fails on replicated data file

A call to **PUTDODA()** in standalone mode on a replicated data file was failing with error **REPL_ERR**. The internal logic has been corrected to resolve this issue.

3.34 Rebuild Error 62 after Low-Level index delete when using KEEPOPEN_LIST

When using `KEEPOPEN_LIST` in *ctsrvr.cfg* to cause data and index files to be kept open by FairCom Server after the application closed them, the following sequence of calls caused the rebuild to fail with error 62 (**LERR_ERR**, file must be opened exclusively):

1. Open data and index files in shared mode.
2. Close data and index files (`KEEPOPEN_LIST` keeps them open).
3. Open index low-level in exclusive mode.
4. Delete index low-level.
5. Open data file using OPNIFIL() without index.
6. Close the data file using CLIFIL (`KEEPOPEN_LIST` keeps the data file open).
7. Rebuild indexes fails with error 62.

The logic has been modified to correct this problem.

3.35 Rebuild with UNIFRMAT enabled - Deleted or resource records treated as active

After rebuilding indexes using a c-tree library with UNIFRMAT enabled, deleted or resource records were marked active if the file was fixed-length and the first field of the record schema was affected by byte ordering. The logic has been modified to correct this.

3.36 r-tree Server crash running concurrent reports

Running concurrent r-tree reports was crashing the server. An internal r-tree global variable was modified during the parsing of r-tree scripts without proper thread synchronization. The global variable was converted to a local variable to avoid concurrent modifications.

All Rights Reserved

3.37 r-tree report causes c-treeACE Server to terminate with Internal Error 7497

An r-tree report on a data file whose DODA only defined fields on part of the fixed-length portion of the data record caused c-treeACE Server to terminate with internal error **7497**. The logic has been modified to allocate a sufficient record buffer regardless of the DODA.

3.38 Stop server initialization immediately when license file initialization fails

When c-treeACE Server was started with no license file and the ADVANCED_ENCRYPTION option was present in *ctsrvr.cfg*, it terminated with a message indicating that the server was not licensed to use this option, rather than with a license error. The logic has been modified correct this behavior.

3.39 Transaction commit fails with error 42 when file opened twice with COMPATIBILITY MULTIOPN_DIFUSR

The transaction commit was failing with error **42** when a connection opened a file twice and the configuration option COMPATIBILITY MULTIOPN_DIFUSR was used. When c-treeACE Server's COMPATIBILITY MULTIOPN_DIFUSR configuration option was used and a connection opened a file twice and updated a record in the file, the transaction commit could fail with a **Record Lock Error** (42). The logic has been modified to address this issue.

3.40 Unexpected old transaction log causes c-treeACE Server to shut down

When c-treeACE Server created a new active transaction log, if a file by the same name existed and a file by that name with an FCQ extension existed, c-treeACE Server terminated with a log write error. The same error occurred if KEEP_LOGS was used and renaming a log file to an inactive log file (FCA extension) failed because an old inactive log with that name existed.

To prevent unexpected server shutdowns, an attempt is now made to first rename old log files if at all possible. The file rename operation will be logged in *CTSTATUS.FCS* with a message similar to:

```
Mon Mar 04 15:39:51 2013
 - User# 00018 Could not rename active log to FCA: attempting to rename FCA file: 67
Mon Mar 04 15:39:51 2013
 - User# 00018 L0000102.FCA
Mon Mar 04 15:39:51 2013
 - User# 00018 ZFD15CA6
```

3.41 VSS Writer does not support 32-bit writer on 64-bit Windows Vista or later

Microsoft documents that Windows Vista and later only support running native VSS writers. For example, on a 64-bit system running Windows Vista or later, only a 64-bit VSS writer is supported.

When the 32-bit c-treeACE Server ran on a 64-bit Windows Vista or newer system (such as 64-bit Windows Server 2008 or Windows 7), the server successfully loaded and initialized the VSS writer DLL. However, the command `vssadmin list writers` did not show an entry for `c-treeACEVSSWriter` and Windows backup did not invoke the VSS writer.

The 32-bit c-treeACE Server now checks before it loads the VSS writer DLL for a 64-bit system running Windows Vista or later. If this is the case, the server logs **error 995** to *CTSTATUS.FCS* and fails its startup:

```
VSS Init: Failed to initialize the 32-bit VSS writer. This version of Windows only supports native 64-bit VSS writers.
Could not initialize server. Error: 995
```

Error 995 is a new error indicating that VSS writer initialization failed.

3.42 Zero log-entry file handle during auto recovery following file compact operation

A message was appearing in *CTSTATUS.FCS* during automatic recovery after FairCom Server was shut down improperly when performing a file compact operation:

```
 - User# 00001 WARNING: zero log-entry file handle (tfil)
Tue Jan 29 11:26:39 2013
 - User# 00001 Dumped stack for server process 4496, log=1, loc=64, rc=0
Tue Jan 29 11:26:39 2013
 - User# 00001 O1 M0 L64 F-4 P0x (recur #1) (uerr_cod=0)
```

The logic has been changed to prevent this message from being erroneously written to the log.

3.43 Memory Leaks

The issues involving memory leaks listed in this section have been addressed.

ctadmn memory leaks fixed

Testing revealed that **ctadmn** was leaking a small amount of memory. The logic has been modified to correct this.

CTDB .NET - GetFieldValue() memory overwrite

An apparent memory leak and a handled "First change exception" was reported in the debugger when calling **GetFieldValue(***IntPtr,int,StringBuilder***)**. A similar problem could occur with the **GetFieldAsString()** function. The logic has been corrected to eliminate this problem.

All Rights Reserved

4. FairCom DB API API Fixes

4.1 CTDB - ctdbAlterTable cannot alter segmented files

Attempting to use **ctdbAlterTable** to alter segmented files that did not actually have any segments other than the "host" was failing with **CTDDRET_NOTSUPPORTED**. The logic has been changed to allow such files to be altered. The alter table code also converts the table into a non-segmented table. It can later be converted back to a segmented file if necessary. There has been no change in the behavior of **ctdbAlterTable** with segmented files that have segments in addition to "host."

4.2 CTDB - ctdbAlterTable did not keep field alignment

The wrong data alignment was seen in a table after altering it. The table creation logic has been improved to use the same alignment as the original table.

4.3 CTDB - ctdbAlterTable did not move COBOL and VTABLE resources to new table

After running **ctdbAlterTable(***..TRUNCATE..***)** on the MRT host, the final result did not contain the MRT resources. The logic has been modified to copy VTABLE and COBOL resources.

4.4 CTDB - ctdbAlterTable (truncate) may change file mode

The **ctdbAlterTable (***truncate***)** function was found to change the file mode when the file name matched a server configuration file list criteria. If *ctsrvr.cfg* contained a line similar to `COMPRESS_FILE *.dat`, a table that originally was not compressed was now compressed after truncating. The logic has been changed to correct this behavior.

4.5 CTDB - ctdbAlterTable (truncate) may lose the VLENGTH property

After using truncate, a VLENGTH table that was created through ISAM became FIXED length. The logic in **ctdbAlterTable(***..TRUNCATE..***)** has been modified to maintain the original VLENGTH mode during a truncate operation.

All Rights Reserved

4.6 CTDB - ctdbAlterTable with CTDB_ALTER_OWNER fails on table not under transaction control

Error -17062 (c-tree error **LERR_ERR**) was seen when running CREATE INDEX on a table that did not support transaction processing. The logic was corrected to add a specific check for the CTDB_ALTER_OWNER case.

4.7 CTDB - ctdbDateTimeSetDate invalid timestamp

If a timestamp initialized to zero was passed to **ctdbDateTimeSetDate()**, the function was failing with an **Invalid Timestamp** (4041) error. The logic in **ctdbDateTimeSetDate()** has been modified to set the Time portion to midnight in such cases without returning an error.

4.8 CTDB - ctdbGetFieldAsSigned on CT_?FLOAT memory overwrite

The **ctdbGetFieldAsString** function used by ISAM Explorer to visualize floating point fields was overflowing a buffer in isolated cases. To correct this, the default float string format was changed to use the shorter representation: 6.953223e-310 or 0.000000:

- 6.953223e-310: Scientific notation (mantissa/exponent), lowercase
- 0.000000: Decimal floating point, lowercase

The side effect of this change is that now **ctdbGetFieldAsString()** may return a floating point using the scientific notation instead of the normal number representation.

It is possible to restore the original behavior by calling **ctdbSetDefFloatFormat(***hSession,"0.000000"***)** after allocating the session.

4.9 CTDB - ctsqlimp Error 101 unlinking after updating server to V10

A **ctsqlimp error 101** was seen while unlinking after updating the server to V10. Tables could not be unlinked if they had been linked with V9 when an owner (*-o* option) had been specified. The logic has been changed to correct this.

4.10 CTDB - ctdbStringToNumber stack corruption causes server crash

When the **ctdbStringToNumber()** function (a FairCom DB API function used in COBOL-to-SQL data conversion) was passed a string that was longer than its 128-byte buffer, it was writing over the end of the 128-byte buffer, corrupting the stack, and causing a crash. A check has been

All Rights Reserved

added to avoid going beyond the boundaries of buffer. When the boundary is reached, **ctdbStringToNumber()** now returns **CTDBRET_INVNUMBER**.

4.11 CTDB - Error 12 in ctdbAddTableXtd with file with no extension and hint set to EXT_PRESENT_P

ctutil *-sqlize* returned error 12 if the table name did not have an extension (e.g. vcusti). The logic has been changed to correct this.

4.12 CTDB - Infinite loop in ctdbCloseAll when ctdbCloseTable failed with network error

If **ctdbCloseAll()** was called and **ctdbCloseTable()** failed with a network error before calling **_ctdbListDeleteEx()**, **ctdbCloseAll()** could get into an infinite loop. The logic has been modified to correct this.

4.13 CTDB - Handling of non-default field delimiters and padding bytes

FairCom DB API has been updated to correctly handle non-default field delimiters and padding bytes.

4.14 CTDB - SetFieldAsBigint may fail with OVERFLOW error when setting CT_INT4U fields

An overflow error was seen when importing a table with a 4-byte unsigned field into SQL and promoting the unsigned fields BIGINTs in SQL. This error may have been seen when inserting or updating values which are beyond the upper limit of a 4-byte signed field.

The **ctdbSetFieldAsBigint()** function called when the actual field in the table is a CT_INT4U, calls **ctdbBigIntToLong()** which returns a signed 4-byte integer.

This situation has been corrected by implementing a new FairCom DB API function, **ctdbBigIntToULong**, which returns an unsigned long (4-byte integer):

ctdbBigIntToULong(*value***)**

- *value* [IN]
- pLong [OUT]
- Returns **CTDBRET_OK** on success)

This function is now used in **ctdbSetFieldAsBigint()** to return an unsigned long when converting a 4-byte unsigned field.

All Rights Reserved

4.15 CTDB - sqlimp does not remove owner prefix from SQL table names

After altering a table with c-treeACE ISAM Explorer, the table name seen in SQL changed from "name" to "owner_name." The logic has been changed to correct this situation.

4.16 CTDB - unlink fails for MRT tables when using prefix

It was found that **unlink** was failing for MRT tables when using prefixes. Error 4131 (**CTDBRET_VTABLEEXIST**) was returned when running the following command:

```
ctutil -sqlize .. -prefix=XX_
```

This was due to logic that was unnecessarily changing MRT table names when the names were extracted from the dictionaries. The logic has been changed so that the MRT table names are not altered during **unlink** when the table names are extracted from the dictionaries.

4.17 CTDB - Wrong field offsets in variable portion of the record

A situation was encountered in which the wrong data was retrieved. In particular, the data looked "shifted." The problem occurred if there was a field that started in the fixed portion of the record and ended in the variable portion followed by other fields.

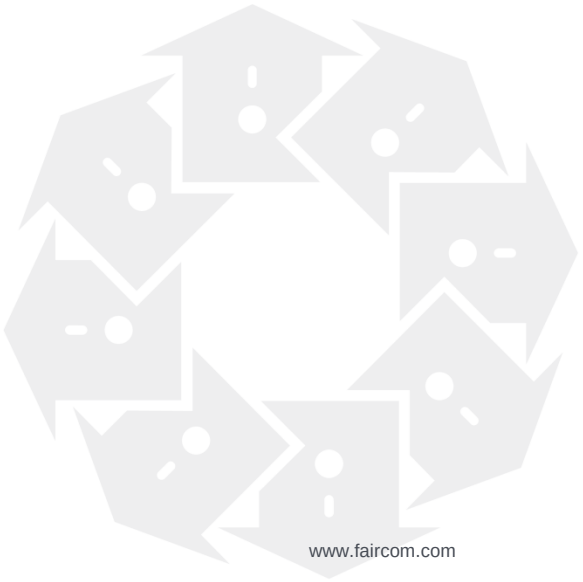
The logic has been corrected as follows:

1. **_ctdbInitRecord()** was modified so that, if there is a field starting in the fixed portion but terminating in the variable portion, that field is considered the first variable field.
2. **_ctdbUpdateRecord()** was modified so that, when starting to analyze the variable portion of the record, it uses the offset of the first variable field as the starting offset.

All Rights Reserved

5. c-treeACE SQL Updates and Corrections

In this release, the SQL interface was the recipient of numerous improvements, which are listed in this section.



5.1 c-treeACE SQL on big-endian systems considers index for backward scan

On AIX systems (or any big-endian system) c-treeACE SQL was previously not considering an index as an option for a backward scan. The logic has been modified to correct this situation. This change can provide performance gains for selected queries where a backward scan will prevent a full table scan from being performed.

5.2 c-treeACE SQL Server memory leak when SQL client establishes GUEST connection

When a V10 SQL client connected as GUEST, a memory allocation was leaked. The logic has been modified to correctly handle a SQL GUEST user.

5.3 c-treeACE SQL Server memory leaks

A few additional isolated memory leaks were discovered within the c-treeACE SQL Server engine and have been corrected with this release.

5.4 c-treeACE SQL Server's fc_get_server_version stored procedure leaks memory when it fails

Memory use was found to increase when a call to c-treeACE SQL Server's **fc_get_server_version** stored procedure failed. The logic has been modified to prevent this from occurring.

5.5 c-treeACE SQL Server process does not appear in Unix process list

On Unix systems, the c-treeACE SQL Server process, **ctreesql**, did not appear in the process list in the shell in which it was started. (On some systems this is always true; on other systems this symptom is only observed when **ctreesql** is run from a shell script.) The logic has been modified to correct this.

All Rights Reserved

5.6 c-treeACE SQL Server terminates when connection limit causes non-secure SQL connection to fail

c-treeACE SQL Server was terminating with an unhandled exception when a non-secure SQL connection attempt failed due to reaching the connection limit. When c-treeACE SQL Server's license allowed logons that are not secure, if a non-secure SQL connection failed because the connection limit was exceeded, the next non-secure SQL connection attempt may have caused c-treeACE SQL Server to terminate with an unhandled exception. The logic has been modified to correct this.

5.7 c-treeACE SQL Server terminates with unhandled exception on starting up

When c-treeACE SQL Server started up, an unhandled exception was sometimes seen. The call stack showed the exception was in **cttcre()**. This exception only occurred on systems that raise an exception when an unaligned memory dereference occured (e.g., Solaris SPARC). The logic has been modified to correct this.

5.8 c-treeACE SQL Server unhandled exceptions when using the SQL Debug keyword, TPESQLDBG=YYYYYYY

During internal testing with TPESQLDBG set to YYYYYYY in *ctsrvr.cfg*, cases were found in which unhandled exceptions occurred. The logic was changed to resolve the exceptions.

5.9 c-treeACE SQL Server unhandled exception creating table with CHECK clause without column name

When a table was created with a CHECK clause and that clause did not contain a column name, c-treeACE SQL Server terminated with an unhandled exception. The following example reproduces this exception. Note that this is not a syntactically correct statement, as the CHECK clause must contain a column reference:

```
CREATE TABLE t (i INTEGER CHECK(1 > 1))
```

The logic has been changed to return a syntax error in this situation.

All Rights Reserved

5.10 ctsqlimp does not mark tables linked specifying a primary key

The **ctsqlimp** tool was not identifying the primary key on tables when using the *-m* switch to specify the primary index. The logic has been corrected.

5.11 Direct SQL - Function added to client API

Testing with Python identified areas in which **executemany** could be improved to properly handle update statements. A new function was added to the client-side C API used in DSQL (ctsqlapi) to expose the server-side **executebatch** logic: **ctsqlExecutebatch**

5.12 Python batch updates with executemany

Testing using Python identified an area in which **executemany** could be improved to work properly with update statements. Using **ctsqlExecute** (DSQL API) to execute a batch with update statements (instead of insert statements) did not return any error but only the first update was actually applied.

Logic was added in the **ctsqlapi ctsqlExecute** function so that it now calls **dh_executebach** when it detects a batched operation. This function can handle both batched inserts and updates. The logic also properly counts updated records (considering that an update can touch multiple rows).

5.13 SQL - 64-bit Windows clients terminate with unhandled exception when connecting to c-treeACE SQL Server on same machine

When a 64-bit Windows SQL client using the C interface (for example, ISQL or the c-treeSQL ODBC Driver) connected to the c-treeACE SQL Server on the same machine, an unhandled exception could occur if the shared memory connection pointer exceeded 4GB of memory. The logic has been changed to correct this.

5.14 SQL - ADO.Net inefficient LVB retrieval - Wrong server version checking

The ADO.Net driver retrieved LVB from the server in chunks of 8192 bytes maximum instead of using a larger buffer, which would have been more efficient. The logic has been changed to improve this.

All Rights Reserved

The logic has also been changed to retrieve the server version numbers (by calling the **fc_get_server_version** stored procedure) the first time the version number is required.

5.15 SQL - Allow ":param is null" syntax

The syntax ":parameter is null" was not supported by our grammar. Changes have been made to allow this syntax. It is now possible to run queries such as:

```
SELECT WHERE (databaseField is null and :ReportParameter is null or (databaseField = :ReportParameter)).
```

5.16 SQL - Attempting to terminate a SQL connection has no effect when user name contains lowercase characters

When a SQL client connected using a user name that contains lowercase characters, an attempt to terminate that connection (using the **ctadmn** utility for example) had no effect.

The function that terminates the connection has been modified to correct this.

5.17 SQL - sqlapi support for batch updates: SQL_ERR_BADSQLDA

Stored procedure execution was failing with error **SQL_ERR_BADSQLDA**. One additional column was recently allocated in the sqlda to return the batch execution result. In a specific server-side check done for stored procedures, the number of columns allocated did not match the number of parameters. The logic has been modified to correct this.

5.18 SQL - Callback library causes crash or terr(7495)

A callback library caused a crash or an AIX terr(7495) when the following SQL statement was executed:

```
select * from vhmain where vhmain.lpnum like 'ABC'
```

The callback library may have been modifying the field and key sizes during table open and it was not taking into account the value of ikeydup when setting the key length. This issue has been corrected.

5.19 SQL - Calling UDF with extra arguments crashes server

Calling a user-defined function (UDF) using parameters and passing more arguments than required crashed the server. Changes have been made to the logic to correct this.

All Rights Reserved

5.20 SQL - Cannot kill server on Unix after first SQL connection occurred

With the proper configuration keyword on Unix, it is possible to shut down the server by hitting **Ctrl+C** or running `kill <server pid>`. However, these techniques did not work after the first SQL connection occurred. The logic has been modified to correct this.

5.21 SQL - Clients fail to connect with error -17012 when PRIME_CACHE and PRIME_INDEX options match names of FAIRCOM.FCS members

With either of the following keywords set, SQL connections failed with **Error 12**, but nothing showed up in *CTSTATUS* from LOWL_FILE_IO:

```
PRIME_CACHE *.dat#6 MB
PRIME_INDEX *.idx#3 MB#0
```

However, if the SQL system tables were opened from ISAM, they could be read, but the following *CTSTATUS* entries were logged:

```
Wed Jul 17 16:33:24 2013
 - User# 00022 Zero node request (ctgetnod)...
Wed Jul 17 16:33:24 2013
 - User# 00022 .\ctreeSQL.dbs\SQL_SYS\ctreeSQL.fdd!syscharstat.idx
```

The logic has been changed to correct this.

5.22 SQL - Crash after failed memory allocation

A failed memory allocation that would have exceeded the server keyword (`TOT_MEMORY`) was causing the server to crash. The logic has been corrected to alleviate this problem.

Note: `TOT_MEMORY` is a deprecated keyword. FairCom does not recommend setting a total memory limit. It was previously recommend ONLY in extreme situations where memory was very limited. This keyword not recommended in production settings.

5.23 SQL - CREATE TABLE AS SELECT unexpected errors

A query similar to the following was failing with **error 20090** even if an alias had been provided for the "upper(tbl)" expression:

```
create table pippo2 as select tbl,upper(tbl) as up from systables where user='admin';
```

The logic has been updated to correct this situation.

All Rights Reserved

A situation was addressed in which comments in the select statement in a CREATE TABLE AS SELECT caused an infinite loop that would never return even if the query by itself took only a few seconds.

Additional changes were made to address other errors encountered when using CREATE TABLE AS SELECT.

5.24 SQL - ctsqlapi functions exported with c++ mangled names on Unix

The *libctsqlapi.so* library was exporting the **ctsqlSetNumericParameterAsString** and **ctsqlGetNumericAsString** functions with mangled C++ names instead of plain C names, which are consistent with the other public functions. The Python interface looks for plain C names. The *libctsqlapi.so* library has been changed to correct this.

5.25 SQL - ctSQLImportTable Error 101 while unlinking after updating server to V10

After updating the server to V10, **ctSQLImportTable()** failed to unlink the table when the actual name starts with `owner_`. The logic has been changed to correct this. If you specify your own **sqllinkcallback** function, contact FairCom to find out what to do to avoid **error 101** on **ctSQLImportTable()** if someone uses `owner_` in their table names.

5.26 SQL - Database conversion fails when REPLICATE matches superfile

SQL database conversion was found to fail when the `REPLICATE` keyword in *ctsrvr.cfg* matched a superfile member name (we do not support replicating superfile members). The `REPLICATE` keyword was changed to ignore superfile members.

5.27 SQL - dbload handling of table with reserved names

Running **dbload** and trying to load data in a table in which the name matched a reserved keyword resulted in an error such as:

```
FATAL : Table link not found

Exiting !!!
```

The logic has been modified to correct this situation.

5.28 SQL - Error -17004 on Delete using rowid as search criteria on tables with foreign keys

Error -17004 (c-tree **KDEL_ERR** error) was seen while executing a delete statement that made explicit use of rowid. The logic has been changed to correctly handle this situation.

5.29 SQL - Failed ALTER TABLE caused server hangs or crash

When the SQL client disconnected after a failed ALTER TABLE, it entered an infinite loop because the table handle had not been removed from the database's table list, causing it to keep looping on that entry. The logic has been updated to correct this.

5.30 SQL - Group By Order By memory leak slowing performance

A memory leak in Group By Order By clauses was slowing performance. The logic has been corrected to remove the leak.

5.31 SQL - Group By with parameters now fails during prepare

A Group By clause with parameters now fails during prepare. The use of parameters in a Group By clause is not supported, however the error handling was not consistent across the various APIs. The logic has been changed to correct this inconsistency by returning error -20127 (**SQL_ERR_BADPARAM**) at query parsing time.

5.32 SQL - Incorrect conversion of BIGINT values on HIGH_LOW systems

In rare circumstances, BIGINT values were being incorrectly converted on HIGH_LOW systems. This could result in a negative value being returned for a BIGINT on a HIGH_LOW system. The logic has been changed to correct this.

All Rights Reserved

5.33 SQL - Incorrect conversion of double values greater than 10^16 on Linux

On Linux systems, c-treeACE SQL Server returns the wrong value when converting double values larger than 10^16. The logic has been changed so that the correct values are now returned.

5.34 SQL - ISQL on Windows takes a long time to fail to connect when server is not running

On Windows, ISQL takes a long time (15 seconds) to fail to connect when the server is not running (or if the wrong port or hostname is specified). The situation has now been rectified to return more quickly. This change should affect ODBC, SQL command line tools, PHP, ctsqlapi and any other API on top of ctsql (Phyton, dbx, etc.).

5.35 SQL - ISQL "Tuple size too high" error on query returning large number of columns

ISQL was generating a "Tuple size too high" error when running a query that returned a large number of columns. The logic has been modified to resolve this issue.

5.36 SQL - Java stored procedures fail to compile on Unix

When writing the *.java* source file used for compiling, the **fwrite()** was sometimes called with a '\0' character as the last character in the stream. This trailing null generated an error from the Java compiler:

```
Error : -20141    Error Description : error in compiling the stored procedure :08: illegal character: \0
```

We now use **fputs()** on Unix systems which avoids this problem by stopping at a '\0' character.

5.37 SQL - Java user.dir using absolute path for JDK 1.7

Whenever the stored procedure dealt with Date, Datetime, or Time types, the procedures failed with one of the following exceptions:

All Rights Reserved

```
java.lang.AssertionError: Default directory is not an absolute path (just the first time)

java.lang.NoClassDefFoundError: Could not initialize class sun.util.calendar.ZoneInfoFile (always but the
first time)
```

The second exception is a consequence of the first exception, and the first exception is due to the *user.dir* Java System property being set to a relative path. On Windows (but not on other platforms), the *user.dir* property was explicitly set using the LOCAL_DIRECTORY setting, which usually is set using a relative path. The logic has been modified to convert the relative path into an absolute path before using it to set the *user.dir* property on Windows.

5.38 SQL - JDBC batch insert inserts incorrect values

The logic has been modified to correct these two errors:

1. When inserting ascending integers into a unique index, every other **executeBatch()** call generated a **KDUP_ERR**.
2. Batch execution failures were seen with a "**Not Enough Parameter**" error although all parameters where properly set (some to NULL).

5.39 SQL - JDBC wrong hard-coded field-size limit

The JDBC driver contained some hard-coded field-size limits that were set too low (2000). This was causing the Java-based GUI tools to truncate strings at 2000 characters. The limits have been changed to correct this problem.

5.40 SQL - JDBC Shared Memory fix

When the *ctSHMjni.dll* was not available and we tried to connect more than once, the shared memory transport class failed with a **NoClassDefFoundError** exception on *ctSHMjni*. The logic has been modified to correct this.

5.41 SQL - JVM memory consumption

Server memory usage was found to increase due to JVM memory usage. The logic has been modified to provide better handling of local references. Some global variables were removed and others were changed to be local to their functions. A number of places where Java exceptions were caught are now properly cleared.

5.42 SQL - Length scalar function may corrupt data

The length scalar function removed trailing spaces from the original data before calculating the length (the length is expected to not count trailing spaces). The logic has been changed so that the original data is not affected by the length function.

All Rights Reserved

5.43 SQL - MONEY problems with precision > 18 digits

MONEY types with precision greater than 18 digits are mapped into CT_NUMBER instead of being mapped into CT_CURRENCY. When using CT_CURRENCY, only 2 decimal digits are stored; when using CT_NUMBER more than 2 decimal digits may be stored. When assigning a numeric to a MONEY type, the logic performs a "rounding."

In the case of negative numbers, the decimal value returned was not "normalized" leading to the error.

Unexpected FairCom DB API overflow errors were seen when setting the MONEY type with the number of digits greater than 19 (the MONEY type has a maximum precision of 32 digits). Also, Select returned unexpected negative values when the Insert inserted positive values.

The logic has been updated to normalize the decimal values after making it negative.

5.44 SQL - Order By ignored in sub-queries

Some queries containing Order By in a sub-query returned incorrect results. The logic has been modified to correct this.

5.45 SQL - Order By with parameters may cause server crash with ODBC

The use of parameters in the Order By clause is not supported. The error handling has been improved to make it consistent across the various API. The error -20127 (**SQL_ERR_BADPARAM**) is returned at query parsing time.

5.46 SQL - Panic running sqlqa

A Select subquery with multiple blocks failed when it referred to a column of outermost block inside the innermost block. A panic situation was logged in *CTSTATUS.FCS*. The logic has been updated to correct this situation.

5.47 SQL - Panic situation using Alias and OR operator

A query that had `table_alias.*` in the select list and OR predicates was causing the server to panic. The logic has been changed to correct this situation.

All Rights Reserved

5.48 SQL - Prevent crash in SQLSnapshot:UpdateSnapshot

A server crash was observed in **SQLSnapshot::UpdateSnapshot**. If another thread called **SQLSnapshot::UnRegister()** (during a SQL disconnect) during **UpdateSnapshot**, then the **ctdbListCount()** was incorrect and **ctdbListItem()** returned a NULL pointer. The logic has been changed to correct this.

5.49 SQL - Query Builder bug with tables with '-' in table name

The SQL Query Builder application was not properly handling the "_" in field names for COBOL files. The issue is actually on field names with an with embedded "-" (dash rather than underscore). The problem was that table names were surrounded with double quotes everywhere in the generated statement except in the JOIN part. The logic has been corrected to add double quotation marks to the JOIN part of the statement.

5.50 SQL - Query containing scalar subquery with outer references returns incorrect NULL column values

Certain SQL queries were returning rows in which some columns erroneously contained NULLs. When c-treeACE SQL is run with pushdown expression support disabled (by adding SQL_OPTION NO_PUSHDOWN to *ctsrvr.cfg*), the query returns the expected results. The logic has been changed to correct the erroneous NULLs.

5.51 SQL - Query with all outer joins crashes server

A SQL query with all outer joins was causing the server to crash. This was caused by a loop that optimizes join order, which was ending with no node pair chosen when the query contained all outer joins. The logic has been changed so that the pair of nodes 0,1 is chosen as the least cardinality pair.

5.52 SQL - Query with large number (>500) of OR predicates and/or large number (>500) of parameters crashes server

A SQL query with a large number (>500) of OR predicates and/or a large number (>500) of parameters was crashing the server. The logic has been changed to correct this.

All Rights Reserved

5.53 SQL - Restrictions not applied to views

A query on a view ran slowly or quickly depending on how the WHERE clause was written. This behavior was observed even if the queries were semantically identical. The logic has been modified to correct this behavior.

5.54 SQL - Second execution of a certain query crashes server

A query that had data-independent scalar functions with an arithmetic operator was crashing the server when it was run for the second time. The following is an example of the type of query:

```
SELECT 1 from syscalctable where SYSTIME>SYSTIME-1800000;
```

The logic has been updated to properly handle this situation.

5.55 SQL - Select with two levels of nested scalar subqueries

The wrong query result was returned from a Select with two levels of nested scalar sub-queries with aggregates. The error was generated by a statement with a non-GroupBy expression in the main Select and a GroupBy expression in a scalar sub-query Select. The logic has been updated to properly handle this situation.

5.56 SQL - Server crash running parameterized insert statement with LVARCHAR parameter and > 50 parameters

A server crash was experienced when running a parameterized insert statement with LVARCHAR parameter and more than 50 parameters. The logic has been modified to correct this.

5.57 SQL - Server crash calling CtreeSqlDataReader.GetBytes()

The server was crashing when executing **CtreeSqlDataReader.GetBytes()** to retrieve an LVB containing 50,000 bytes. The **data_to_binary** prototype has been changed to avoid the overflow that cause this problem.

All Rights Reserved

5.58 SQL - Exception when a SQL client connected while server was shutting down

On Windows, if a SQL client connected using the shared memory protocol when c-treeACE SQL Server was shutting down, the server terminated with an unhandled exception. This has now been corrected.

5.59 SQL - Specifying multiple options in DH_JVM_OPTION_STRINGS may cause memory overwrite, etc.

The setting specified in DH_JVM_OPTION_STRINGS did not always produce the desired effects. Specifying multiple options sometimes caused memory overwrites and unexpected behavior. The number of options has been increased from 10 to 100 and other changes were made to the logic to correct this.

5.60 SQL - Exception when RPC error occurs during SQL shared memory client logon

On Unix systems, c-treeACE SQL Server was terminating with an unhandled exception when an RPC error occurred during SQL shared memory client logon. The logic has been changed to correct this.

5.61 SQL - Slow performance on Group By Order By

A query containing a Group By and an Order By clause was non-optimal. Different indexes were used to perform the Group By and Order By operations when the same index could have been used for both. Modifications in the logic address this issue and improve the performance.

5.62 SQL - Stored procedures failed with SQL_ERR_NOSTMT

Stored procedures were failing with error 20156, **SQL_ERR_NOSTMT**. The logic has been changed to correct this.

All Rights Reserved

5.63 SQL - Stored procedure not available if called before JVM initialized

If a SQL client attempted to run a Java stored procedure when c-treeACE SQL Server was starting and had not yet initialized the JVM, Java stored procedure support was not available even though it was properly configured. This state persisted for that database until c-treeACE SQL Server was restarted. (A client that connected to a different database after the JVM was initialized was able to use Java stored procedure support.)

A new state variable is now set after the JVM initialization is complete and the SQL shared memory thread waits for that variable to be set before accepting connections.

5.64 SQL - Uninitialized variables cleanup

Occasional JDBC socket I/O exceptions were encountered due to un-initialized data on this stack. The logic has been changed to correct this.

5.65 SQL - User-Defined Functions returned wrong result

An internal issue in the handling of parameters and result values of user-defined functions (UDF) could lead to the wrong result from the UDF. The logic has been modified to correct this situation.

5.66 SQL - Wrong query result joining on rowID with extra conditions

The wrong query result was returned when joining on the rowID and having extra conditions. The logic has been changed to correct this.

5.67 SQL - Wrong query result when mixing inner and outer joins

Queries of the following form were found to return incorrect results if there was a complex join condition or a predicate between tbl2 and tbl3:

```
(tbl1 LOJ tbl2) Inner Join tbl3
```

The logic has been modified to correctly address this situation.

All Rights Reserved

5.68 SQL - Wrong selectivity calculation for unique index with multiple components; last operator Not EQ

The optimization logic was incorrectly calculating the selectivity for a given index with given values (when specified) and a given operator. It contained an optimization for unique indexes when all index components were used and all operators were EQ (since only 1 row matches in this situation). The optimization logic has been modified to correct this situation.

5.69 SQL - MM crash when multiple cursors requiring MM are opened

A crash was sometimes seen when multiple cursors were open at the same time for queries requiring the MM subsystem. The problem did not occur if cursors were properly closed when they were not used anymore. The logic has been changed to continue and return an error code up the stack in such situations.

All Rights Reserved

5.70 SQL Optimizer

SQL - Optimizer now removes redundant outer joins

There are cases where queries may contain useless or redundant pieces. The optimizer identifies redundant predicates. It now identifies situations where an outer join is completely useless and can be removed. This avoids fetching rows from a table and discarding them.

The following is an example of a case where the outer join can be removed:

```
select SUM(b.j) mysum FROM basic100k b LEFT OUTER JOIN rhs1 r ON b.i = r.f4;
```

SQL - Slow query with useless OR predicate

A query containing the condition (field != 1 or field !=2), which always evaluates to TRUE, was found to run very slowly. The OR optimizer tried to optimize the useless OR causing a non-optimal execution plan. The logic has been modified to correct this.

All Rights Reserved

6. ADO.NET

Several improvements have been made to the ADO.NET interface.

All Rights Reserved

6.1 ADO.NET Provider - CommandBehavior.CloseConnection OR-ed with another option

When CommandBehavior.CloseConnection was OR-ed with another CommandBehavior option, the connection stayed open. The logic has been modified to accept OR-ed values.

6.2 ADO.NET Provider - DataReader getValue(...) method returning NULL

The **CtreeSQLDataReader** method **GetValue(...)** should return **System.DBNull** in cases where the value is null. However, for field types of LVarChar and LVarBinary, it returned **NULL**. The logic that checks for null has been modified to correct this situation.

6.3 ADO.NET Provider - Error when reading non-Unicode char fields from Unicode server

c-treeACE SQL Explorer was sometimes seen to return an error or display garbage when using a Unicode-enabled server and creating a table with a non-Unicode character column. When the server is Unicode enabled, columns of type Char must be set to NChar in sqlda types array since they are always treated as NChar even if the table type is Char. This behavior has been corrected.

6.4 ADO.NET Provider - ExecuteScalar() not called with CALL "procedure" syntax

When executed with **ExecuteScalar()**, the (first row) of the Result Set of the Stored Procedure should be returned. Instead, it returned the output parameter, which cannot be cast to (int). This was because, when the statement type was SQL_CALL, the **ExecuteScalar()** function was not called. The logic has been modified to correct this.

6.5 ADO.NET Provider - Numeric types handled incorrectly

When Numeric types were set up as a byte array, the **FcValue.Write()** method was throwing an exception. The logic has been changed to provide a proper conversion.

All Rights Reserved

6.6 ADO.NET Provider - Views with parameters bug

When executing a query similar to `SELECT * from myView` where `myView` was a view that required parameters, the view parameters were not properly evaluated and the command `Prepare` call returned with the number of parameters set to 0. The logic has been modified to correct this issue.

6.7 ADO.NET Provider - "Wrong Number of Parameters" calling stored procedure

The code involving the check for parameter names was causing an error message "**Wrong Number of Parameters**." The logic has been modified to correct this problem.

All Rights Reserved

7. PHP Support

PHP is a useful platform for developing web-based applications. The following changes have been made to the FairCom PHP provider.

All Rights Reserved

7.1 PHP - Invalidate prefetched elements when transaction is ended

When a commit or rollback was executed (either explicitly or implicitly by the auto-commit logic), all open cursors were closed, which caused any further fetch to fail. The PHP interface fetches in groups of 20 (or the value set by `ctsql.rowsperfetch`) which are cached by the client. When the 21st record is needed, the driver sends a request to the server, which returns **error -20040** (cursor not opened). The driver has been modified so that any fetch for opened queries at the time of the transaction close now returns **error -20040** immediately without considering the row already cached.

7.2 PHP - Execution problem on Mac OS X 64-bit

A PHP execution problem on Mac OS X 64-bit caused garbage to be seen on screen when running the tutorials. The logic that identifies 64-bit platforms has been improved to correctly support Mac OS X.

7.3 PHP - Runtime error loading library on recent Linux versions

A PHP runtime error was seen when loading the library on recent Linux versions. The logic has been corrected to eliminate this.

All Rights Reserved

8. Other Improvements

The numerous changes in this release include many improvements listed in this section.

All Rights Reserved

8.1 ctsqlimp: Wrong data in SQL dictionary causes Error -20000 on imported tables with public permissions

Unlinking and re-linking a table with the *-public=ro* option caused an error message "SQL internal error" **-20000** during SQL queries. The logic has been updated to correct this.

8.2 ctrdmp fails with Error 75 in Dynamic Dump with more than 1024 active connections

During its automatic recovery phase, **ctrdmp** was failing with error 75 (**TROW_ERR**, Bad Transaction Owner Error) when a dynamic dump was performed with more than 1024 active connections. The logic has been modified to correct this.

Note: A 32-bit **ctrdmp** could still fail with error 75 if run on transaction logs created by a 64-bit c-treeACE Server, which might support more than 2048 connections.

8.3 Data Compression deactivated for Bound Server Library, UNIX

Data Compression has been deactivated for the Bound Server library in Unix.

Note: If someone wants to use the Bound Server model, they can consider the newer Server DLL model, which includes this feature.

8.4 Error 598 using conditional expression on field in variable-length part of record when file uses custom field delimiter

When a data filter was set on a CT_STRING field in the variable-length portion of a record when the file used a different field delimiter than the default of 0x0, a call to read the record failed with error 598 (**CVAL_ERR**, Could Not Evaluate Conditional Expression). The logic has been corrected to use the delimiter stored in the schema.

8.5 FairComConfig - Error when VS2012 is not installed

The FairComConfig returned an error while configuring the registry settings for Visual Studio if Visual Studio 2012 was not installed. The logic has been changed to correct this.

All Rights Reserved

8.6 File compact fails with Error 944 on compressed file

When the index files did not exist, file compact was failing with **error 944** (Bad augmented record file mode at create) when compacting a compressed data file or enabling compression during the compact. The logic has been changed to correct this issue.

Note: The IFIL path (directory name) adjustment logic from the rebuild utility (**ctrbldif**) was applied to the compact utility (**ctcmpcif**). This logic is useful when the files reside in a different directory than the one that is stored in the data file's IFIL resource and the data and index file paths in the IFIL resource are identical. In this case, if the filename that is passed to the compact utility contains a path, the index file directory names are changed to the specified path.

The **ctinfo** utility was updated to display the ctCompressRec attribute for a file when present in the file's flmode3 header field.

8.7 GTREC - Correct automatic key transformation on index allowing duplicates

GTREC was found to return **Unexpected Record** on an index that allowed duplicates when automatic key transformations were enabled. The logic has been changed to correct this.

8.8 IICT - Fixed potential infinite loop

Very high CPU utilization and a hung thread were found during certain IICTs (Immediate Independent Commit Transactions). The routine that evaluates whether or not a key level lock is associated with an active transaction uses hash bins to match a transaction number with the thread that owns the transaction. In the unlikely situation that both the encompassing transaction and the IICT were assigned to the same hash bin, an infinite loop could occur. The logic has been changed to correctly handle this situation.

8.9 Java - CLASSPATH default setting changed to ":" instead of ";"

The Unix *ctsrvr.cfg* defaults had the wrong delimiter character for the SETENV CLASSPATH. The two jar files are now separated by a colon (“**:**”) rather than a semicolon (“**;**”):

```
; JDK environment settings - Be sure to set the JDK to your version.
;SETENV
CLASSPATH=../Java/jdk1.6.0_33/jre/lib/rt.jar:../bin/ace/sql/classes/ctreeSQLSP.jar
```

This was not a bug *per se* because the entry is commented out, but it may be used as a template for setting up stored procedures.

All Rights Reserved

8.10 Java FairComIsam Library - CtreeSession: Added check for NULL JCtree

The CtreeSession class tried to create a JCtree class but did not check for failures that may result in a null object. A check has been added and an exception is now thrown if the object is null.

8.11 JEE - JDK1.6 compatibility fix

While testing Java Persistence with JEE (Glassfish), it was noticed that FairComCtreeDB_JCA was failing to build when using JDK1.6 version. The logic has been changed to correct this.

8.12 JDBC - Out of Bounds exception in CtreeDA:enlarge()

When adding elements to a batch, an exception occurred with the following trace:

```
java.lang.ArrayIndexOutOfBoundsException
        at java.lang.System.arraycopy(Native Method)
        at ctree.jdbc.CtreeDA.enlarge(CtreeDA.java:813)
        at ctree.jdbc.CtreePreparedStatement.addBatch(CtreePreparedStatement.java:2375)
```

The **enlarge()** function has been changed to correct this error.

8.13 JDBC - Open socket caused client thread to hang after JDBC connection failed

The client thread was hanging because a socket was left open after a JDBC connection failed. The logic has been changed to ensure that the socket is closed after a failed connection.

8.14 JDBC - Shared Memory

When the shared memory DLL was not available and one tried to connect more than once, the shared memory transport class failed with a **NoClassDefFoundError** exception on **ctSHMjni**. On the first attempt to use shared memory, the **UnsatisfiedLinkError** exception was issued; after that, the exception was different. The logic has been modified to correct this.

8.15 Mac OS X - Dependency on CoreFoundation framework when linking c-treeSQL with Java support

In the c-treeACE SQL Server makefile, we added a dependency on the CoreFoundation framework when linking the *ctreedbs* shared library on Mac OS X.

All Rights Reserved

8.16 ODBC - Too many indexes in connection

Creating a table with over 200 columns was causing ODBC error **-20051** (Too many indexes on the table). The limit of 200 columns for all the indexes for the same table has been increased to 250 columns for this customer.

8.17 r-tree causes c-treeACE Server to terminate with Internal Error 7497 if DODA defines fields on fixed-length portion of record

When running an r-tree report, c-treeACE Server terminated with an internal **error 7497** if the data file's DODA only defined fields on part of the fixed-length portion of the data record. The logic has been modified to correct this situation.

8.18 Security - User names with embedded spaces

When connected with a user name that contained a space, an error could be generated because the utility was not surrounding the user name with double quotation marks. The logic has been changed to correct this so that user names with embedded spaces are now handled properly.

8.19 VCL - BLOB field not stored after CopyFrom()

From RAD Studio XE2 to XE3, the Dataset (VCL) framework changed with the result that the BLOB field is not marked for update during the **CopyFrom()** call. Additionally, **Post()** is not checking if there is any BLOB field pending update. The logic has been changed to correct these issues.

All Rights Reserved

8.20 Replication Agent Updates

Enhancements listed in this section have been made to the first member of the FairCom Advanced Module Series: The FairCom Replication Agent

Replication Agent - Avoid duplicate log position entries when automatic recovery reads more than one checkpoint in its scan phase

After c-treeACE Server terminated abnormally and restarted, the replication reader log position list contained two entries having the same Replication Agent name (REPLAGENT). The extra entry caused c-treeACE Server to keep all its transaction logs. The logic has been modified to correct this.

Replication Agent - Improved counting of failed operations

The Replication Agent's counting of operations that fail because the file cannot be opened on a target server has been improved. When applying changes fails because a file cannot be opened, the failed operation count is now incremented so that the transaction gets aborted, and when logging the exception records, the appropriate operation counter is incremented.

Replication Agent - Improved calculation and persistence of replication reader transaction log requirements

Additional improvements were made to calculation and persistence of Replication Agent transaction log:

1. The Replication Agent's call to set the checkpoint position in its log tran list failed with a KDUP_ERR (meaning that an entry for that log already existed, which is not expected). This possibility has been corrected.
2. In some cases the Replication Agent did not know to inform the source server of logs that it needed to keep. For example, when the log scan started before the first checkpoint in a log and there was an active transaction that existed entirely in that log, the Replication Agent thought there was no need to keep previous logs (because there were no pending transactions for previous logs).
3. The source server deleted logs required by replication readers after automatic recovery. The following problems were corrected:
   - The calculation of which logs to keep when the source server started had the sense of the log number comparison reversed so logs needed by replication would not be kept

even though the replication minimum log number state variable was set to the correct value.

- If the server was terminated and restarted after automatic recovery with no additional checkpoint activity in the logs, the old log number was not preserved, causing transaction logs to be orphaned (a cosmetic issue) and causing the replication log requirement check to be skipped.
- When c-treeACE Server keeps more than the minimum number of logs (4) at startup due to replication requirements, the count of active logs was incorrect.
- When a Replication Agent sets its log position it temporarily sets its required log to -1 and if the setting of the position fails, it sets its required log to zero. The original log requirement is now maintained when setting the log position. If the setting of the log position is successful, the new minimum log number is set; otherwise the original log number is maintained.
- When a replication reader registers its log requirement, we now write a checkpoint to the transaction logs so that the log requirement is immediately made persistent.

4. When more than one Replication Agent was starting its scan at the same time, some of them failed to open the previous log with **error 96**. This happened because we used an exclusive mode to open the previous log.
5. Two options were added to the **ctrepd** utility:
   - *-nominlog* disables log persistence on the source server. If this option is used, the source server determines the current log requirement based on which log is being read by the utility. When the option is not used (the default), ctrepd receives REPL_CHKPNT entries, and so this option is useful for seeing those entries.
   - *-unqid*:<id> specifies the replication unique ID to use when using the -minlog option. This is useful if you are running more than one ctrepd utility on the same source server at the same time.
6. We added the command getlogtail to the repadm utility. This option displays replication agent exception log entries starting with the last record in the exception log when repadm is started.

NOTE: The replication agent now stores its current minimum required log in the REPLSTATEDT.FCS record on the target server. This required adding data to the end of the variable-length record. The replication agent automatically converts an existing REPLSTATEDT.FCS file to the new format (adding a new DODA) when it starts.

Replication Agent - Unhandled exception when transaction contains multiple updates

c-treeACE Server was terminating with an unhandled exception when a transaction contained multiple updates to a record and the replication log scan started after some of the updates. The logic has been modified to correct this.

Replication Agent - Unhandled exception if data source and target files use different alignment than agent

The Replication Agent was terminating with an unhandled exception when adding or updating a record on the data target if the data source and target files use a different alignment than the alignment used to compile the agent's c-tree client library.

The logic has been modified to correct this and a new keyword has been added to set the data source alignment. This feature is enabled by specifying the following option in the Replication Agent configuration file, *ctreplagent.cfg*:

```
use_target_alignment yes
```

This option is enabled by default.

Because most applications use the compiler's default alignment (8 bytes), so there is no difference in alignment, this exception is not commonly observed.

All Rights Reserved

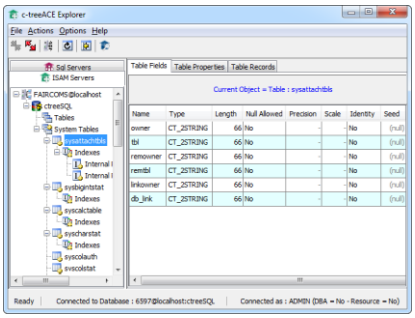
9. GUI Tools

The graphic tools have been refreshed. They are available in Java for cross-platform support. In addition, we have made numerous updates to the original .NET versions of these valuable tools.

9.1 Cross-platform Java tools

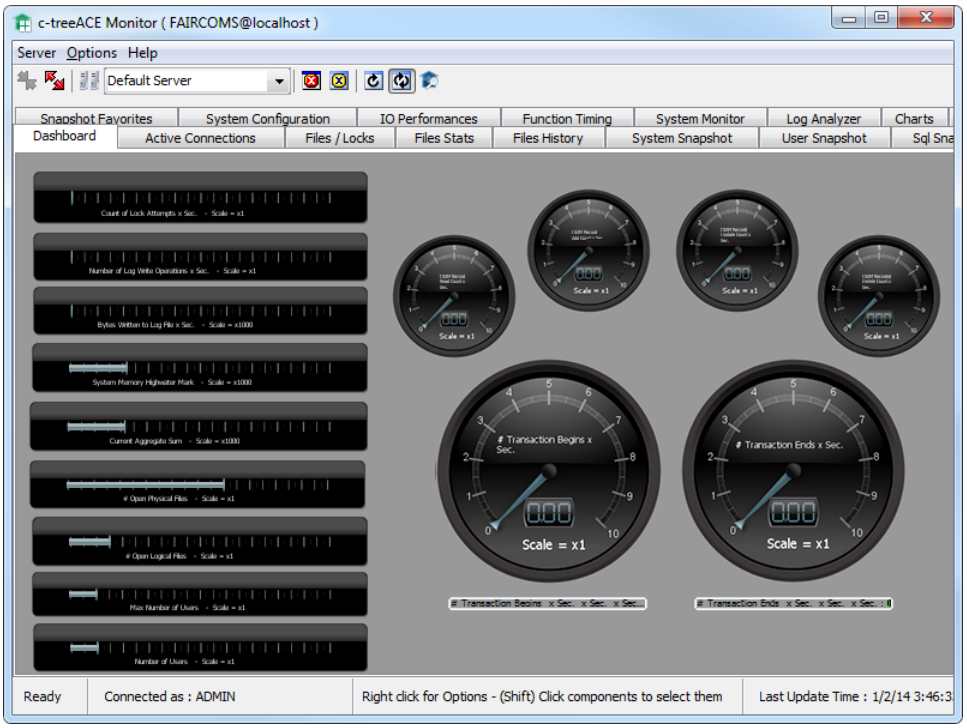
The cross-platform Java versions of the graphical tools have been greatly enhanced. Some of the tools have been combined to simplify their operation. The following tools are available:

c-treeACE Explorer - View and edit both SQL and ISAM databases from a single tool

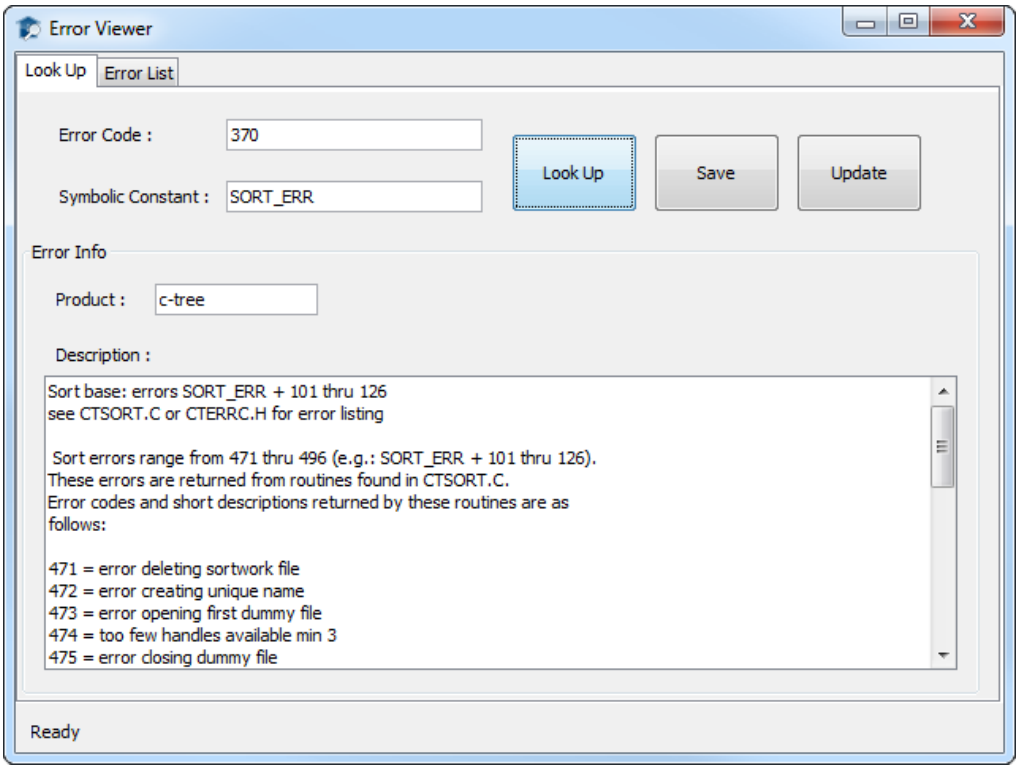


All Rights Reserved

c-treeACE Monitor - Gauges and detailed monitoring of your c-tree databases



Error Viewer - An updated version of the error viewer includes an up-to-date list of c-treeACE errors



All Rights Reserved

All Rights Reserved

9.2 Changes to the .NET GUI Tools

c-treeACE SQL Explorer - Create view with special characters in name

The c-treeACE SQL Explorer was not putting quotations marks around table names that contained special characters (e.g., "−" and "_"), which caused errors when creating views. The logic has been improved to correct this.

c-treeACE SQL Explorer - Empty lines added to script and lost server connection

Two issues were corrected in c-treeACE SQL Explorer:

- c-treeACE SQL Explorer was adding empty lines added to the bottom of the script window.
- If the connection to the server was lost when running a script, the **Run** button was grayed out could not be enabled, even if the connection was reconnected.

c-treeACE SQL Explorer - Errors display in Scripts

When running a script using c-treeACE SQL Explorer, some errors (-17408, -17920, for example) display a dialog box. Some errors, such as -17008, are written to the output pane instead. All errors now go to the output pane.

c-treeACE SQL Explorer - Export Schema bug on "double" return values

When creating SP/UDF or Triggers, any argument or return value of type "double" had to be specified as "double precision" because return values for UDF did not append the "precision" keyword. the logic has been corrected to eliminate this problem.

c-treeACE SQL Explorer - Statements still executed after removing them

Changes to the logic corrects this behavior:

Run c-treeACE SQL Explorer. Open a script, delete the contents from the script window, and click "Run". The commands still run even though they have all been deleted.

c-treeACE SQL Explorer - Wrong formatting of DECIMAL and NUMERIC columns

While viewing records in c-treeACE SQL Explorer, the DECIMAL and NUMERIC columns were properly formatted on the first row only. This has been corrected.

All Rights Reserved

c-treeACE SQL Explorer and c-treeACE Explorer - Comment chars in quotes 'xx--'

c-treeACE SQL Explorer and c-treeACE Explorer were updated to properly handle comment characters in quotation marks.

In a statement similar to the following the −− characters (which usually mean the start of a comment) must be ignored since they are inside quotation marks:

```
select  * from trans_codes where rawtranscode = 'AA--'
```

The tool was not ignoring them and instead was stripping them and running :

```
select  * from trans_codes where rawtranscode = 'AA
```

The logic has been adjusted to properly handle such cases.

c-treeACE SQL Explorer and c-treeACE Explorer - Scripts may lose spaces

The **Scripts** panes in c-treeACE SQL Explorer and c-treeACE Explorer were removing spaces from the script in certain circumstances. If you ran a script with fields containing more than one consecutive space in a string, the actual values inserted had fewer spaces within the string. This occurred when the line did not end with a semicolon and it had more than one space character between words.

For example:

```
insert into trans_codes (rawtranscode, int_type, transcode)
values ('XX   X  X XXX', 28, 'XX   X  X XXX');
```

where:

- rawtranscode - Character(40)
- int_type - Character(3)
- transcode - Character(20)

The script has 3, 2, and 1 spaces between the Xs:

```
XX   X  X X
```

However, it was stored with only one space between the Xs:

```
XX X XX
```

The logic has been changed to correct this.

Dr. c-tree - Delete record bug

The Delete Record function in Dr. c-tree was found to throw an exception if no records were loaded in the grid. This has been corrected.

All Rights Reserved

Dr. c-tree - Error 963 on UpdateDoda

Attempting to update a DATOBJ field type using Dr. c-tree was failing with **error 963**. The logic has been changed to correct this.

Dr. c-tree - Records not read on files with reclen = 0

If a file had a header member reclen == 0 (all fields of type vlength), no record read was performed. The logic has been changed to correct this.

c-treeACE ISAM Explorer - Crash on disconnect

An exception was occurring in c-treeACE ISAM Explorer when opening the database table (-adv mode) and viewing the records, then selecting the Disconnect button.

The logic has been corrected to test for a closed connection condition when trying to validate a record row.

c-treeACE ISAM Explorer - 32-bit Windows version now runs on Windows XP

Changes have been made to c-treeACE ISAM Explorer for 32-bit Windows so that it now runs on Windows XP.

c-treeACE Monitor - Error 516 (ctNOGLOBALS Not Allocated)

Running c-treeACE Monitor, switching to the System Monitor page, and clicking Start Monitor returns **Error 516** (ctNOGLOBALS Not Allocated). The logic has been changed to address this issue.

c-treeACE Monitor - Exception on tool start

In rare conditions the c-treeACE Monitor tool was throwing an exception on startup. the logic has been modified to correct this.

c-treeACE Monitor - Exception when server stops

c-treeACE Monitor was causing an exception when the server stopped. Enabling the function timings and then stopping the server caused an exception on every autoupdate. The logic has been changed to correct this.

All Rights Reserved

10. Notable Compatibility Changes

This section lists c-treeACE changes that affect compatibility. It is important to review these issues to ensure that your product functions correctly after upgrading to this version.

10.1 Camouflage Data Protection (CAMO) not available in c-treeACE Express

Camouflage Data Protection (CAMO) provides protection from casual observation by masking (scrambling) the protected data. This support has been disabled in c-treeACE Express version. CAMO or "Camouflage" is an older, legacy method of hiding data, which is not a standards-conforming encryption scheme, such as AES. It is not intended as a replacement for Advanced Encryption or other security systems.

10.2 Encryption of c-treeACE SQL System Tables

c-treeACE SQL System Tables are now encrypted by default to protect database meta data. The encryption algorithm is AES with a 128-bit key when advanced encryption is in use. Data is hidden with the less-secure CAMO when not using the advanced encryption. CAMO or "Camouflage" is an older, legacy method of hiding data, which is not a standards-conforming encryption scheme, such as AES. It is not intended as a replacement for Advanced Encryption or other security systems.

All Rights Reserved

10.3 File rebuild memory limit - File memory usage keyword for rebuild memory usage

Support was added for `SORT_MEMORY` to specify the maximum memory to use in a rebuild. The `SORT_MEMORY` keyword specifies a memory size in bytes and can use the MB and GB keywords (unlike `MAX_K_TO_USE`).

The maximum `SORT_MEMORY` value is:

- 4 TB - 1 for 64-bit c-treeACE
- 4 GB - 1 for 32-bit c-treeACE

As the `SORT_MEMORY` option is more intuitive, its use is recommended over MAX_K_TO_USE. (MAX_K_TO_USE remains available for backward compatibility). If both SORT_MEMORY and MAX_K_TO_USE are specified in *ctsrvr.cfg*, only the one that is specified last in the configuration file takes effect.

10.4 Dynamic Dump of Non-Transaction Files Defaults to YES

Dumping non-transaction files is a popular option, and as a consequence, the default for `PERMIN_NONTRAN_DUMP` has been reversed and now defaults to YES.

10.5 Suppress logging File Delete Error for I0000000.FCS during startup

c-treeACE Server no longer logs the following error message to *CTSTATUS.FCS* when it starts, as this is considered a normal condition:

```
Wed Jan  9 10:44:05 2013
- User# 00001  Scanning transaction logs
Wed Jan  9 10:44:05 2013
- User# 00001  idltfil: File <I0000000.FCS> unlink failed with error={2}
```

10.6 ODBC SQLColAttributes renamed

SQLColAttributes() is a deprecated ODBC 2.0 function replaced by **SQLColAttribute()** in ODBC 3.0. Internally, our ODBC driver used a function named **SQLColAttributes** to implement **SQLColAttribute()**. On Unix, this function (and all global functions) were exported by our *libctodbc.so*. When an ODBC 2.0 application called **SQLColAttributes()**, the driver manager was supposed to convert this to **SQLColAttribute()** for an ODBC 3.0 driver, but unixODBC used **SQLColAttributes()** if it existed.

This revision renames our **SQLColAttributes()** to **iSQLColAttributes()** to avoid this confusion.

All Rights Reserved

Note: On Windows, **SQLColAttributes** was not exported by the ODBC driver.

10.7 ctOpenSequence() returns NO_ERROR when specified sequence does not exist

A call to the c-treeACE ISAM API **ctOpenSequence()** specifying the name of a sequence that does not exist returns **NO_ERROR**. The logic has been corrected so that such a call now returns **SEQNAM_ERR** (as documented).

10.8 ctSETENCRYPT - Passing a NULL to disable encryption

Passing a NULL value for the *key* parameter to ctSETENCRYPT was inadvertently resetting *keylen* to 0, which was disabling encryption. In V10.3 and later, the code in the client library interface to **ctSETENCRYPT()** has been changed so that it is consistent with the following definition:

In the **ctSETENCRYPT()** function, *keylen* > 0 enables encryption for all files created after that point, and *keylen* <= 0 disables encryption. This is true regardless of whether *key* is NULL or not.

Note: A non-NULL *key* is *required* when a call to **ctSETENCRYPT()** is made that enables basic encryption; *key* is *ignored* when using Advanced Encryption.

10.9 Rebuild Fails with Error 484 (Could Not Open Sort Work File)

When `COMPRESS_FILE` was specified in *ctsrvr.cfg* with a filename specification that matched rebuild sort work file names (for example `COMPRESS_FILE *.*` to match all filenames), a rebuild failed with **error 484** (could not open sort work file).

NOTE: **Error 401** could occur when opening any data file (not just a rebuild sort work file) that is created with resource support disabled and whose name matches the `COMPRESS_FILE` filename specification.

The logic has been changed so that the `COMPRESS_FILE` keyword does not enable data compression for a data file that is created with resource support disabled.

All Rights Reserved

10.10 Treat fixed-length compressed data files consistently across batch record returns, inserts, and updates

Prior to c-treeACE V10.3, a fixed-length data file that had record compression turned on was implemented internally as a variable-length file with the **ctAugmentedFxd** file attribute. The **ctAugmentedFxd** attribute enforced the constraint that all the records are of the specified fixed length. After compression, the records may be of different lengths. To determine if your file is behaving in this manner, you can execute the **ctinfo** utility over the data file and look for the presence of the **ctAugmentedFxd** file mode in the c-tree extended header block.

ctBEHAV_BAT_FXDCMP changes the behavior of the batch retrieval and insertion code so that compressed, fixed-length data file records are now treated as fixed-length records, not variable-length. This new behavior is consistent with other c-tree API calls and the BAT_UPD option. **ctBEHAV_BAT_FXDCMP** is on by default, but `COMPATIBILITY NO_BAT_FXDCMP` placed in the *ctsrvr.cfg* file will turn off the new behavior at runtime.

The new behavior affects BAT_RET_REC and BAT_RET_BLK retrieval operations by removing the record size field that precedes each record image.

BAT_RET_FIX is mutually exclusive with BAT_RET_REC, and designed to be used with variable-length files, and not used in conjunction with BAT_RET_BLK.

Before the new behavior, BAT_RET_FIX would have resulted in compressed, fixed-length data file records being returned like variable-length records. Now they will be returned like fixed-length records.

The new behavior affects the BAT_INS option by expecting the input buffer to be in fixed-record length format for compressed fixed-length data file records.

All Rights Reserved

10.11 Unix Shared Memory Protocol Not Freeing Shared Memory Segments (different client and server user accounts)

When clients used the shared memory protocol on Unix systems and the client and server processes were run under different user accounts, shared memory segments could be left behind after the connections were closed. The **ipcs** *-m* listing showed shared memory segments with no processes attached. In V10.3 and later, the logic has been modified to correct this.

Note: These changes add a field to the client and server logon data structures to pass the user ID between the client and server processes, resulting in the following compatibility considerations:

An old client can connect to a new server and it will behave as it did before these changes.

A new client cannot connect to an old server using shared memory. It will receive **error 133** if the server is configured to only use shared memory. It will connect with TCP/IP if the server is using both shared memory and TCP/IP. The old server will log the following message to *CTSTATUS.FCS*:

```
FSHAREMM: The client's shared memory version (2) is not compatible with the
server's shared memory version (1)
```

At startup, the FairCom Server now logs messages to *CTSTATUS.FCS* to indicate the shared memory directory used for logon purposes and the shared memory protocol version that it is using:

```
FSHAREMM: SHMEM_DIRECTORY=/tmp/ctreedbs/
FSHAREMM: Protocol version=2
```

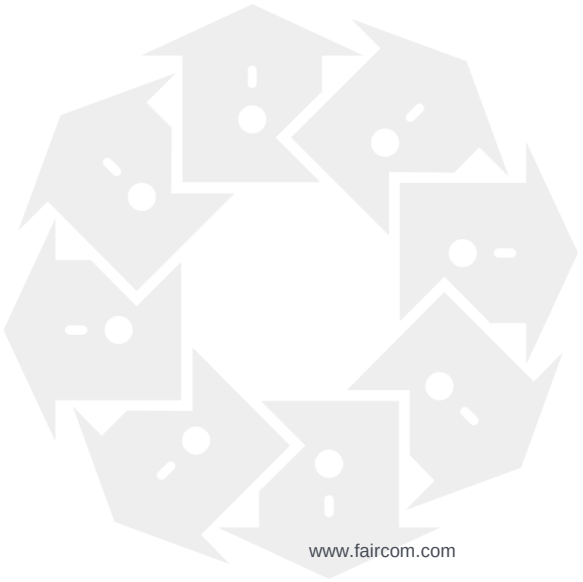
All Rights Reserved

11. FairCom Typographical Conventions

Before you begin using this guide, be sure to review the relevant terms and typographical conventions used in the documentation.

The following formatted items identify special information.

Formatting convention	Type of Information
Bold	Used to emphasize a point or for variable expressions such as parameters
CAPITALS	Names of keys on the keyboard. For example, SHIFT, CTRL, or ALT+F4
FairCom Terminology	FairCom technology term
FunctionName()	c-treeACE Function name
Parameter	c-treeACE Function Parameter
`Code Example`	Code example or Command line usage
utility	c-treeACE executable or utility
filename	c-treeACE file or path name
`CONFIGURATION KEYWORD`	c-treeACE Configuration Keyword
CTREE_ERR	c-treeACE Error Code



All Rights Reserved

Copyright Notice

Copyright © 1992, -2025 FairCom USA Corporation. All rights reserved.

No part of this publication may be stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording or otherwise without the prior written permission of FairCom USA Corporation. Printed in the United States of America.

Information in this document is subject to change without notice.

Trademarks

FairCom DB, FairCom EDGE, c-treeRTG, c-treeACE, c-treeAMS, c-treeEDGE, c-tree Plus, c-tree, r-tree, FairCom, and FairCom's circular disc logo are trademarks of FairCom USA, registered in the United States and other countries.

The following are third-party trademarks: Btrieve is a registered trademark of Actian Corporation. Amazon Web Services, the "Powered by AWS" logo, and AWS are trademarks of Amazon.com, Inc. or its affiliates in the United States and/or other countries. AMD and AMD Opteron are trademarks of Advanced Micro Devices, Inc. Macintosh, Mac, Mac OS, and Xcode are trademarks of Apple Inc., registered in the U.S. and other countries. Embarcadero, the Embarcadero Technologies logos and all other Embarcadero Technologies product or service names are trademarks, service marks, and/or registered trademarks of Embarcadero Technologies, Inc. and are protected by the laws of the United States and other countries. HP and HP-UX are registered trademarks of the Hewlett-Packard Company. AIX, IBM, POWER6, POWER7, POWER8, POWER9, POWER10 and pSeries are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. Intel, Intel Core, Itanium, Pentium and Xeon are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries. ACUCOBOL-GT, Micro Focus, RM/COBOL, and Visual COBOL are trademarks or registered trademarks of Micro Focus (IP) Limited or its subsidiaries in the United Kingdom, United States and other countries. Microsoft, the .NET logo, the Windows logo, Access, Excel, SQL Server, Visual Basic, Visual C++, Visual C#, Visual Studio, Windows, Windows Server, and Windows Vista are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. Oracle and Java are registered trademarks of Oracle and/or its affiliates. QNX and Neutrino are registered trademarks of QNX Software Systems Ltd. in certain jurisdictions. CentOS, Red Hat, and the Shadow Man logo are registered trademarks of Red Hat, Inc. in the United States and other countries, used with permission. SAP® Business Objects, SAP® Crystal Reports and SAP® BusinessObjects™ Web Intelligence® as well as their respective logos are trademarks or registered trademarks of SAP. SUSE" and the SUSE logo are trademarks of SUSE LLC or its subsidiaries or affiliates. UNIX and UNIXWARE are registered trademarks of The Open Group in the United States and other countries. Linux is a trademark of Linus Torvalds in the United States, other countries, or both. Python and PyCon are trademarks or registered trademarks of the Python Software Foundation. isCOBOL and Veryant are trademarks or registered trademarks of Veryant in the United States and other countries. OpenServer is a trademark or registered trademark of Xinuos, Inc. in the U.S.A. and other countries. Unicode and the Unicode Logo are registered trademarks of Unicode, Inc. in the United States and other countries.

All other trademarks, trade names, company names, product names, and registered trademarks are the property of their respective holders.

Portions Copyright © 1991-2016 Unicode, Inc. All rights reserved.

Portions Copyright © 1998-2016 The OpenSSL Project. All rights reserved. This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (http://www.openssl.org/).

Portions Copyright © 1995-1998 Eric Young (eay@cryptsoft.com). All rights reserved. This product includes cryptographic software written by Eric Young (eay@cryptsoft.com). This product includes software written by Tim Hudson (tjh@cryptsoft.com).

Portions © 1987-2020 Dharma Systems, Inc. All rights reserved.

This software or web site utilizes or contains material that is © 1994-2007 DUNDAS DATA VISUALIZATION, INC. and its licensors, all rights reserved.

Portions Copyright © 1995-2013 Jean-loup Gailly and Mark Adler.

Portions Copyright © 2009-2012 Eric Haszlakiewicz.

Portions Copyright © 2004, 2005 Metaparadigm Pte Ltd.

Portions Copyright © 2008-2020, Hazelcast, Inc. All Rights Reserved.

Portions Copyright © 2013, 2014 EclipseSource.

Portions Copyright © 1999-2003 The OpenLDAP Foundation.

Open Source Components

Like most software development companies, FairCom uses third-party components to provide some functionality within our technology. Often those third-party components are selected because they are a standard in the industry, they offer specific functionality that is easier to license than to develop and maintain in the long run, or they provide a proven and inexpensive solution to a particular business need. Examples of third-party software FairCom uses are the OpenSSL toolkit that provides Transport Layer Security (TLS) for secure communications and the ICU Unicode libraries to provide wide character support (think international characters and emojis).

Some of these third-party components are the subject to commercial licenses and others are subject to open source licenses. For open source solutions that we incorporate into our technology, we include the package name and associated license in a notice.txt file found in the same directory as the server.

The notice.txt file should always stay in the same directory as the server. This is particularly important in instances where your company has redistribution rights, such as an ISV who duplicates server binaries and (re)distributes those to an eventual end-user at a third-party company. Ensuring that the notice.txt file "travels with" the server binary is important to maintain third-party and FairCom license compliance.

1/30/2025

12. Index

All Rights Reserved

All Rights Reserved

All Rights Reserved

All Rights Reserved