

test link

Release Notes

# FairCom RTG V3 Release Notes

Audience

**Developers**

Subject

**FairCom FairCom RTG V3 Release Notes**



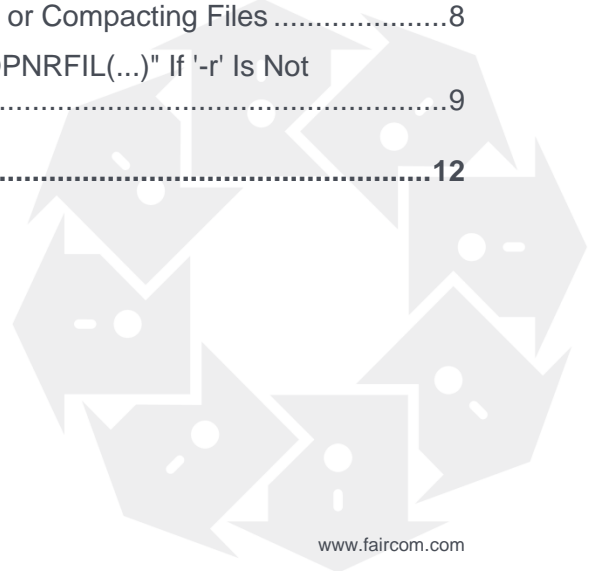
© Copyright 2025, FairCom Corporation. All rights reserved. For full information, see the FairCom Copyright Notice (page x).



FairCom®

# Contents

<b>1.</b>	<b>FairCom RTG V3 Release Notes .....</b>	<b>1</b>
1.1	Fixed Locking Issue Reading Record Updated under Transaction .....	2
1.2	File Structure and Duplicate Scan Problem Fixed .....	2
1.3	Possible Unexpected c-tree Error 57 Deleting a Record (loss of record lock) Fixed .....	3
1.4	OnConvertError "value:?" Correctly Handles UTF-8 Strings.....	3
1.5	Memory Leak Creating Extended Segment Definition Fixed .....	4
1.6	Unexpected Record Returned Reading Key on Corrupted File .....	4
1.7	Fixed Crash at Exit with Server DLL Model.....	4
1.8	ctree.conf Problem Parsing the Priority Attribute Properly.....	4
1.9	<file type> Now Takes Precedence Over <file name>.....	4
1.10	<instance ctshmemdir> to Set Shared Memory Directory under Unix.....	5
1.11	<redirinstance> - Fixed COBOL Error 9A (file locked) with Micro Focus .....	5
1.12	<redirinstance> - COBOL Operation No Longer Falls Back to Default File System if FairCom RTG Cannot Load Library .....	6
1.13	Shared Memory Configuration of FairCom RTG Client .....	6
<b>2.</b>	<b>FairCom RTG Tools &amp; Utilities .....</b>	<b>7</b>
2.1	ctutil -sqlink File Path .....	7
2.2	ctutil -sqlize - sqlimport No Longer Requires Exclusive Access Importing MRT Table .....	7
2.3	ctutil -info Command No Longer Reports non-RTG files as Encrypted .....	7
2.4	ctutil - Prints Error Message in Case of XDD Parsing Error .....	8
2.5	ctutil - Repair Corrupt Data Option Rebuilding or Compacting Files .....	8
2.6	ctmigra - No Longer Logs "ERROR 15:12:2 OPNRFIL(...)" If '-r' Is Not Specified.....	9
<b>3.</b>	<b>Index .....</b>	<b>12</b>



# 1. FairCom RTG V3 Release Notes

*This document lists many fixes that have been made to improve the stability of FairCom RTG.*

*Be sure to also see the FairCom RTG V3 Update Guide (<https://docs.faircom.com/doc/v3-rtg-update-guide/>) for important details, especially the *Compatibility Notes* (<https://docs.faircom.com/doc/v3-rtg-update-guide/CompatibilityNotes.htm>) chapter.*

*Since FairCom RTG is based on FairCom DB at its core, most of the fixes in the FairCom DB V12 Release Notes (<https://docs.faircom.com/doc/v12rel/>) also apply to FairCom RTG V3.*

## 1.1 Fixed Locking Issue Reading Record Updated under Transaction

A program rewrote a record in a transaction then tried to read it with another logical FD. All reads were performed without locking, and the following configuration was active:

- `<detectlock>yes</detectlock>` - To test if read records are locked by someone.
- `<locktype>0</locktype>` - To read the content of locked records.
- `<runitlockdetect anyunlock="yes">no</runitlockdetect>` - To avoid record locking between programs in the same run unit.

In this scenario, the read performed via a separate FD did not retrieve the record content but it also did not get a record locked error. The logic has been modified to correct this.

## 1.2 File Structure and Duplicate Scan Problem Fixed

For physical variable-length tables (for example, enabling compression will force tables to be physically variable-length) with alternate keys allowing duplicates, FairCom RTG creates and includes a unique auto-incrementing value termed as "Serial Segment" (SRLSEG) appended to these keys forcing an internal uniqueness.

Files created with FairCom RTG V11.6 could lack the SRLSEG segment that would have been in their alternate key definitions if created with V11.2.3 and earlier. One symptom is the record length would be one byte larger than with V11.2.3 and earlier.

The following problems could be seen on files created with the affected FairCom RTG builds:

1. A scan on a duplicate key with record updates could touch more records than expected because some records were matched multiple times (the record could be repositioned forward in the physical file on update and so the key gets found again).
2. ISAM applications accessing the file that did not take *RTG!* (<https://docs.faircom.com/doc/ctcobol/ctadrtgi-AddSQLindexesoverexistingCOBOLtables.htm>) resource information into account may not behave properly due to a different record and index layout.

The original behavior has been restored, correcting this issue.

**Note:** Only files created from builds in this date range will be affected. Existing files remained as they were. Any new files created should be identified, inspected and potentially updated.

Note that **ctutil -rebuild** will NOT fix the incorrect definitions.

Although **ctutil -upgrade** can fix an affected file by forcing a full reconstruction of the table, it is difficult to find the files that need to be fixed. For this reason, a utility has been created to find and fix the affected files. See below for this utility.

## 1.3 Possible Unexpected c-tree Error 57 Deleting a Record (loss of record lock) Fixed

When using the FairCom RTG configuration option `<runitlockdetect anyunlock="no">no</runitlockdetect>` (or *iscobol.properties* option `autolock_allowed=1`) and not using COBOL's SELECT option "WITH ROLLBACK", an update on a record that has been locked from a different file handle unexpectedly released the lock causing a delete operation to fail with c-tree error 57 (**DADV\_ERR**, "proper lock not held") because FairCom RTG requires records to be locked in order to delete or modify them.

For example, assume the following sequence:

1. OPEN file USING handle1
2. READ rec1 USING handle1 WITH LOCK
3. OPEN file USING handle2
4. UPDATE rec1 USING handle2
5. DELETE rec1 USING handle1

Step 5 fails with c-tree error 57 because the lock acquired in step 2 has been released unexpectedly in step 4.

The lock is not expected to be released because the option `anyunlock="no"` should permit the lock to be released only from the same file handle. The lock is released by the automatic transaction commit that occurs during the UPDATE operation. The file uses automatic transactions because it has transaction support (c-tree file mode `ctPREIMG`) but it has been opened without the "WITH ROLLBACK" option.

The automatic transaction commit by default releases all the locks acquired during the transaction. However, it also releases the locks acquired on the same record using a different file handle.

This behavior has been corrected.

## 1.4 OnConvertError "value:?" Correctly Handles UTF-8 Strings

When using the *onConvertError* attribute in an XFD, an attempt to use `onCovertError="value: something"` failed to trigger an error with a null string, giving the impression the logic worked, when it really failed. The logic performing the conversions from UTF-8 to UTF-16 has been modified to take possible invalid UTF strings into account and to fail with an error message.

## 1.5 Memory Leak Creating Extended Segment Definition Fixed

A memory leak was seen when creating an extended KSEGDEF index (this type of index is used only by FairCom RTG, Unicode, and JSON). The **PUTKSEGDEF(ctKSEGcreidx)** function leaked memory from a temporary buffer on success. The logic has been modified to correct this.

## 1.6 Unexpected Record Returned Reading Key on Corrupted File

A COBOL READ KEY operation on a corrupted index file returned a different record than the given key. The logic has been modified to correct this.

## 1.7 Fixed Crash at Exit with Server DLL Model

An isCOBOL program configured to run with the server DLL model (set environment "ctree.bound\_server" to "1") could crash in **ctputmem()** at exit time. The logic has been modified to correct this.

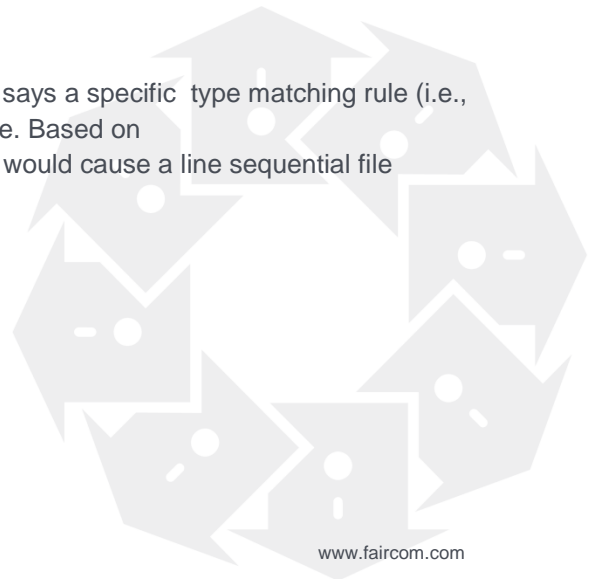
## 1.8 ctree.conf Problem Parsing the Priority Attribute Properly

Setting the `<file priority>` to a negative value (other than -1) or to a non-numeric string like `<file priority='acb' />` resulted in the priority being set to -1. The logic has been modified to correct this.

## 1.9 <file type> Now Takes Precedence Over <file name>

The FairCom RTG `<file type>` keyword did not take precedence over `<file name>` as stated in documentation.

The documentation, *File Matching Precedence* ([https://docs.faircom.com/doc/ctcobol/\\_file\\_.htm#o71510](https://docs.faircom.com/doc/ctcobol/_file_.htm#o71510)), says a specific type matching rule (i.e., I, R, L, or S) takes precedence over the \* (unspecified) type. Based on that rule, it is expected that the following configuration file would cause a line sequential file named *lineseq.txt* to be handled by `<redirinstance>`:



```

<config>
  <redirinstance lib="libcobrts64.so" func="EXTFH"> <!--Micro Focus-->
    <file type="R"/> <!--relative files-->
    <file type="S"/> <!--sequential files-->
    <file type="L"/> <!--line sequential files-->
  </redirinstance>
  <instance> <!--ctreeRTG-->
    <file/> <!--indexed files-->
    <file name="lineseq.txt"/>
  </instance>
</config>

```

Instead, the program failed because the line sequential file was handled by `<instance>`. Because priorities were not specified in the above `ctree.conf`, they were assigned automatically by FairCom RTG leading to unexpected results.

The logic has been modified to set priorities to -32767 only for the "name=\* dir=\* type=\*" rule (which is the default file rule) and assign 0 to all others. Now, rules that specify only the file type (if different than "\*") are given a priority of 0, as documented.

## 1.10 `<instance ctshmemdir>` to Set Shared Memory Directory under Unix

On Unix platforms, the FairCom RTG server shares a directory with the FairCom RTG clients when they communicate via the shared memory protocol. It is possible to configure the FairCom RTG server to change the default shared memory directory with configuration keyword `SHMEM_DIRECTORY`. Using different shared memory directories allows running multiple FairCom RTG servers with the same name on the same machine. To configure the FairCom RTG clients to set the shared memory directory, it is necessary to set the c-tree global variable `ctshmemdir`.

This revision implements a new `<instance ctshmemdir>` attribute to configure the shared memory directory for the `<instance>` by setting the c-tree global variable `ctshmemdir`.

The `<instance ctshmemdir>` is ignored under Windows as it is meaningful only for Unix platforms.

## 1.11 `<redirinstance>` - Fixed COBOL Error 9A (file locked) with Micro Focus

A COBOL program that attempted to create a line-sequential file failed with error **9A** (file locked) if the `DYNREDIR` environment variable was set to `CTEXTFH` library and `<redirinstance>` was configured to use Micro Focus library `MFFH` (`<redirinstance lib="cobmffh64.3.so" func="MFFH">`).

This is a complex situation involving a loop created by the redirection and `CTEXTFH` and `MFFH` libraries. To resolve this situation, the `MFFH` library no longer manages the operation so that it is handed to the next file handler and eventually to the default Micro Focus file handler.

## 1.12 <redirinstance> - COBOL Operation No Longer Falls Back to Default File System if FairCom RTG Cannot Load Library

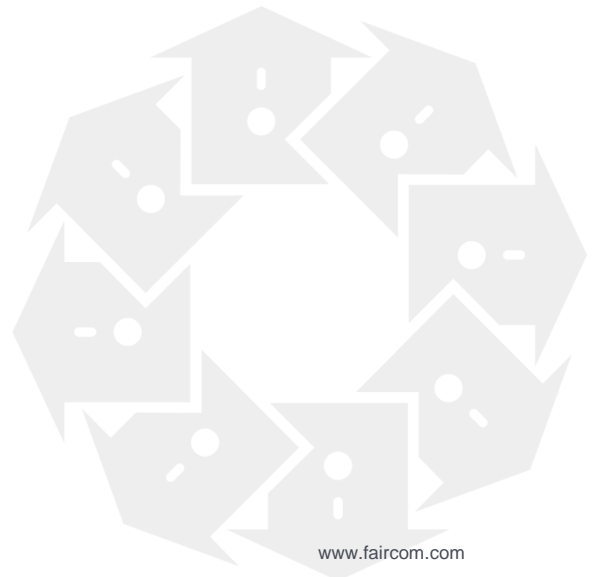
When a <redirinstance> was specified with the wrong `lib=` and/or `func=` attribute in the FairCom RTG configuration file, FairCom RTG did not fail with an error. Instead, it fell back to the default file system, which handled the COBOL operation that was supposed to be handled by FairCom RTG.

The **CTEXTFH()** function has been changed to return a value of 1 in case of a configuration/switcher error. This tells Micro Focus that the file could belong to FairCom RTG, causing it to fail instead of retrying with the next file system. This new behavior logs an error in the FairCom RTG log, which should help you to fix the <redirinstance> attributes.

## 1.13 Shared Memory Configuration of FairCom RTG Client

On Unix platforms, the FairCom RTG server shares a directory with the FairCom RTG clients when they communicate via the shared memory protocol. It is possible to configure the FairCom RTG server to change the default shared memory directory with configuration keyword `SHMEM_DIRECTORY`. Using different shared memory directories allows running multiple FairCom RTG servers with the same name on the same machine.

To configure the FairCom RTG clients to set the shared memory directory, a new <instance `ctshmemdir`> attribute has been added to configure the shared memory directory for the <instance> keyword. The <instance `ctshmemdir`> keyword is ignored under Windows as it is meaningful only for Unix platforms.





## 2. FairCom RTG Tools & Utilities

### 2.1 ctutil -sqlink File Path

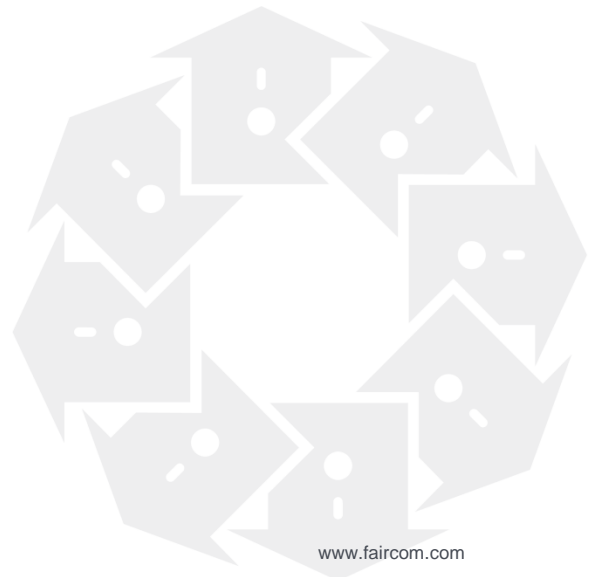
In FairCom RTG, the **ctutil -sqlink** file path was prefixed with '.' or './' if the path separator differed from the platform's path separator. The logic has been modified to accommodate both native and foreign path separators ('\ in Windows and '/' in Unix/Linux), which solves this problem.

### 2.2 ctutil -sqlize - sqlimport No Longer Requires Exclusive Access Importing MRT Table

**ctutil -sqlize** failed with internal error **19:-8** when file pool was effective. The problem has been fixed.

### 2.3 ctutil -info Command No Longer Reports non-RTG files as Encrypted

The FairCom RTG **ctutil -info** command could report non-RTG files as encrypted. The logic has been modified to correct this problem.





## 2.4 ctutil - Prints Error Message in Case of XDD Parsing Error

When **ctutil** finds a problem with an attribute (or in general any problem) it generates an error message that helps identifying the issue. In certain situations, this error message was lost.

```
ctutil -sqlcheck myfile myfile.xdd
```

Previously, when **ctutil** encountered an error parsing or interpreting the fields' information, it failed to provide information about the problem. For example, the following points to an error in the XDD, but provides no additional help to identify the problem:

```
ctutil -sqlcheckmyfile myfile.xdd

ctutil Version 12.0.0.92-190908
Initialized from (ctree.conf)
Error opening/parsing file myfile.xdd
```

**ctutil** has been modified to provide additional information, such as:

```
Initialized from (ctree.conf)
[GetXDDinfo] {ctxmlGetFIELDLIST} Field attribute julianBase='16001231' near line 16 is invalid
Error opening/parsing file stckhd.xdd
Parameter not correct.
```

## 2.5 ctutil - Repair Corrupt Data Option Rebuilding or Compacting Files

A previous change in the c-tree rebuild/compact logic caused these operations to return error **DCPT\_ERR** (1107, data file corrupt, restore to backup) if an invalid variable length c-tree record header was detected in the data file. A *-repairdata* command-line option was added to **ctrblidif** and **ctcmpcif** utilities to force the rebuild or compact operation to attempt scanning for a valid record header, repair the invalid header, and continue the operation.

This modification introduces the *-repairdata* option for **ctutil -rebuild** and *-compact* commands to force rebuild or compact to attempt a repair if damaged data records are detected. Upon error 1107 **DCPT\_ERR**, **ctutil** now outputs the following error message:

```
Damaged data records detected. Restore data file from backup or use -repairdata and attempt to restore to a functional state.
```



## 2.6 **ctmigra - No Longer Logs "ERROR 15:12:2 OPNRFIL(...)" If '-r' Is Not Specified**

Log message "ERROR 15:12:2 OPNRFIL(...)" was displayed if **ctmigra** option `-l` was enabled and option `-r` was not specified. If option `-r` was not specified, **ctmigra** attempted to open the destination file to check if it already exists and then errors out. If the destination file did not exist, the FairCom RTG file-open logic logged an error indicating that the file was missing.

The logic has been modified so that, if `-l` is enabled and no `-r` option is used, the **ctmigra** auto-generated FairCom RTG configuration file now includes a `<log><error><missingfile>no</missingfile></error></log>` rule for the destination file so that no error is logged in case of a missing file.

# 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 Haszlkiewicz.

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

## 3. Index

<	
<file type> Now Takes Precedence Over <file name>.....	4
<instance ctshmemdir> to Set Shared Memory Directory under Unix.....	5
<redirinstance> - COBOL Operation No Longer Falls Back to Default File System if FairCom RTG Cannot Load Library.....	6
<redirinstance> - Fixed COBOL Error 9A (file locked) with Micro Focus.....	5
<b>C</b>	
Copyright Notice.....	x
ctmigra - No Longer Logs.....	9
ctree.conf Problem Parsing the Priority Attribute Properly.....	4
ctutil - Prints Error Message in Case of XDD Parsing Error.....	8
ctutil - Repair Corrupt Data Option Rebuilding or Compacting Files.....	8
ctutil -info Command No Longer Reports non-RTG files as Encrypted.....	7
ctutil -sqlize - sqlimport No Longer Requires Exclusive Access Importing MRT Table.....	7
ctutil -sqlink File Path.....	7
<b>F</b>	
FairCom RTG Tools & Utilities.....	7
FairCom RTG V3 Release Notes.....	1
File Structure and Duplicate Scan Problem Fixed.....	2
Fixed Crash at Exit with Server DLL Model.....	4
Fixed Locking Issue Reading Record Updated under Transaction.....	2
<b>M</b>	
Memory Leak Creating Extended Segment Definition Fixed.....	4
<b>O</b>	
OnConvertError.....	3
<b>P</b>	
Possible Unexpected c-tree Error 57 Deleting a Record (loss of record lock) Fixed.....	3
<b>S</b>	
Shared Memory Configuration of FairCom RTG Client.....	6
<b>U</b>	
Unexpected Record Returned Reading Key on Corrupted File.....	4

