FairCom White Paper

End User Deployment Recommendations

Contents

c-treeACE Server End User Best Practices

The following best practices should be communicated to all technical and development teams working with c-treeACE. It is also suggested this information should be appropriately included with product documentation as part of deployed installations.

Published 2/20/2018

1.1 Restrict Access to c-treeACE Server Files

The c-treeACE Server engine requires exclusive access to the files under its control. Therefore do not attempt to copy, delete or in any way replace a file being controlled by the c-treeACE Server engine while the Server is running. This can lead to data corruption and Server instability.

1.2 Automatic Recovery

Never interrupt the c-treeACE process during automatic recovery. Data and indexes will be left in an unknown state, resulting in probable data loss. This situation may require restoring data from a clean backup to ensure absolute data integrity.

File ID Overflow during Recovery

When a transaction controlled file is opened by the server, it is assigned a unique *fileid* which is used to reference it in the transaction logs. This number is stored as a 4 byte integer. If large numbers of files are opened and closed repeatedly, available file numbers can be rapidly consumed , leading to a "Pending File ID overflow" message to be logged. This message is logged once every 10,000 file opens. The *fileid* can be reset by shutting down the server cleanly (so no recovery is needed), and removing existing transaction logs (*L*.FCS* and *S000*.FCS*).

An **FUSE_ERR** error (22) during recovery indicates that the FILES setting should be increased to complete recovery. A RECOVER_FILES keyword (NO by default) is also available that controls this specifically.

```
RECOVER_FILES <number of files | NO>
```

> **Note:** Recent improvements to only assign the *fileid* when a file is actually updated, and not just opened and read from, can substantially reduce the file numbers consumed.

1.3 External Third-Party Utilities

c-treeACE server technology requires that it is the only process operating over any data and index files under its control. Third-party backup (and some virus scan) utilities should never touch any files while the c-tree Server has the files open. Doing so on the transaction logs or on a transaction controlled file may terminate the server process. When c-treeACE encounters a write error on these

types of files it can no longer guarantee integrity or recovery and must terminate to protect the state of the current system.

If a third-party backup is desired, either use the c-tree Dynamic Dump backup facility to create a copy of the files and point the third-party backup to the resulting backup file, or shut down the c-tree Server. and then perform the backup.

It is also possible to use a disk-level copy as an acceptable backup strategy, provided c-treeACE is put into a quiet state (using either “Quiesce” or File Block) before the hardware-level copy.

c-treeACE V9.2 and later for Windows also includes Windows VSS support (Volume Shadow Service) for enhanced integration with third-party backup tools.

1.4 File Rebuilds

In the event a problem does occur and requires files to be rebuilt, it is possible to rebuild the files using either a stand-alone utility, or through the c-treeACE Server. To avoid the potential for wrong PAGE_SIZE settings, and creating indexes without the *TRNLOG* file mode, FairCom recommends performing all rebuilds through the c-treeACE Server.

- The c-treeACE V9 Server includes a new keyword, `MAX_K_TO_USE`, controlling the amount of memory utilized for file rebuilds. The default is 64 KB. Increasing this value to as large as practical (not starving the operating system for memory) can provide faster rebuilds than using single-user rebuild utilities.
- The stand-alone utility has traditionally been a faster method to rebuild files, however, it is important to ensure that the index page size is set appropriately (using the PAGE_SIZE switch). The c-treeACE Server defaults to 8192 byte page sizes (64 sectors - 128 bytes/sector), while c-tree standalone technology defaults to 2048 bytes (16 sectors). The stand-alone rebuild utility should also be built with c-tree Transaction Processing Logic enabled such that files are created with the appropriate *TRNLOG* file mode.

1.5 Java Requirements for c-treeACE SQL

Here are the Java requirements for past and current versions of c-treeACE SQL. Remember, Java is used for Stored Procedures, Triggers, and User Defined Functions. If the Java runtime engine (JRE) is not available, then these features are unsupported. The Java SDK is required to compile the source Java modules prior to running.

c-treeACE SQL Version	Java Version Required
11	1.6 or later
10.3	1.6
10.0	1.6
V9.3+	1.6
V9.2	1.5
V9.1	1.5
V9.0	1.5
V8.27	1.5
V8.14	1.4 (1.3 on AIX)

c-treeACE Server Configuration Recommendations

The following options can be defined in the c-treeACE Server configuration file *ctsrvr.cfg* and should be taken into consideration for all new and existing c-treeACE customers:

2.1 SKIP_MISSING_FILES

Do not operate with `SKIP_MISSING_FILES` enabled in the *ctsrvr.cfg* configuration file by default. This configuration may cause files to be inadvertently skipped during recovery, making it difficult to restore the database back to a consistent state. This keyword should only be utilized when confirmed necessary (by reviewing specific error messages in the *CTSTATUS.FCS* status log file), and then promptly removing it. (Commenting the keyword in the configuration file is suggested, as it is then readily available if absolutely needed.)

2.2 KEEP_LOGS

To take fullest advantage of c-tree's recovery options, it is advisable to keep as many transaction logs as practical -- up through the last two backups if possible. The c-tree restore process can start with an existing backup and roll forward through remaining logs if available, restoring data to the latest possible time point.

2.3 FORCE_LOGIDX

FairCom has a feature that reduces the amount of time it takes for automatic recovery to complete, especially in systems with high transaction rates. By adding the keyword `FORCE_LOGIDX ON` to your *ctsrvr.cfg* file, the c-tree Server will store a few additional bytes of information per index node within the c-treeACE transaction logs, which will greatly reduce the amount of time automatic recovery will take. There is no performance loss with the production system for enabling this feature and you do not need to rebuild indexes to enable this feature.

`FORCE_LOGIDX` supports these settings:

- ON forces all indices to use *LOGIDX* entries.
- OFF forces all indices not to use *LOGIDX* entries.
- NO uses existing file modes to control *LOGIDX* entries.

Note: *LOGIDX* is ON by default with c-treeACE V9.2 and later.

Index