NAME

ost::SharedFile −

This class defines a database I/O file service that can be shared by multiple processes.

SYNOPSIS

```
#include <file.h>
```

Inherits **ost::RandomFile**.

Public Member Functions

SharedFile (const char *path)
Open or create a new database file.
SharedFile (const **SharedFile** &file)
Create a shared file as a duplicate of an existing shared file.
virtual **~SharedFile** ()
Close and finish a database file.
Error restart (void)
Restart an existing database; close and re-open.
Error fetch (char *address=NULL, **ccxx_size_t** length=0, off_t position=-1)
Lock and Fetch a portion of the file into physical memory.
Error update (char *address=NULL, **ccxx_size_t** length=0, off_t position=-1)
Update a portion of a file from physical memory.
Error clear (**ccxx_size_t** length=0, off_t pos=-1)
Clear a lock held from a previous fetch operation without updating.
Error append (char *address=NULL, **ccxx_size_t** length=0)
Add new data to the end of the file.
off_t **getPosition** (void)
Fetch the current file position marker for this thread.
bool **operator++** (void)
bool **operator--** (void)

Detailed Description

This class defines a database I/O file service that can be shared by multiple processes.

Each thread should access a dup of the database object, and mutex locks can be used to preserve transaction integrety if multiple threads are used.

SharedFile is used when a database may be shared between multiple processes. **SharedFile** automatically applies low level byte-range 'file locks', and provides an interface to fetch and release byte-range locked portions of a file.

Author:

David Sugar <dyfet AT ostel DOT com> This class defines a database I/O file service that can be shared by multiple processes.

Constructor & Destructor Documentation

ost::SharedFile::SharedFile (const char * path)

Open or create a new database file. You should also use Initial.

Parameters:

path pathname of database to open.

ost::SharedFile::SharedFile (const SharedFile & file)

Create a shared file as a duplicate of an existing shared file. **Parameters:**

file original file.

virtual ost::SharedFile::~SharedFile () `[virtual]`

Close and finish a database file.

Member Function Documentation

Error ost::SharedFile::append (char * address = `NULL`, ccxx_size_t length = `0`)

Add new data to the end of the file. Locks file during append.

Parameters:

address address to use, or NULL if same as last I/O.
length length to use, or 0 if same as last I/O.

Error ost::SharedFile::clear (ccxx_size_t length = `0`, off_t pos = `-1`)

Clear a lock held from a previous fetch operation without updating. **Returns:**
errSuccess on success.

Parameters:
length length to use, or 0 if same as last I/O.
pos file position to use or -1 if same as last I/O.

Error ost::SharedFile::fetch (char * address = `NULL`, ccxx_size_t length = `0`, off_t position = `-1`)

Lock and Fetch a portion of the file into physical memory. This can use state information to fetch the current record multiple times.

Returns:
errSuccess on success.

Parameters:
address address to use, or NULL if same as last I/O.
length length to use, or 0 if same as last I/O.
position file position to use -1 if same as last I/O.

off_t ost::SharedFile::getPosition (void)

Fetch the current file position marker for this thread. **Returns:**
file position offset.

bool ost::SharedFile::operator++ (void)
bool ost::SharedFile::operator-- (void)
Error ost::SharedFile::restart (void) `[inline, virtual]`

Restart an existing database; close and re-open. **Returns:**
errSuccess if successful.

Reimplemented from **ost::RandomFile**.

Error ost::SharedFile::update (char * address = `NULL`, ccxx_size_t length = `0`, off_t position = `-1`)

Update a portion of a file from physical memory. This can use state information to commit the last read record. The current lock is also cleared.

Returns:
errSuccess on success.

Parameters:
address address to use, or NULL if same as last I/O.
length length to use, or 0 if same as last I/O.
position file position to use or -1 if same as last I/O.

Author

Generated automatically by Doxygen for GNU CommonC++ from the source code.