Java NIO – Memory-Mapped Files with

ry-Mapped Files
Buffer

If you know **how java IO works at lower level**, then you will be aware of buffer handling, memory paging and other such concepts. For conventional file I/O, in which user processes issue read() and write() system calls to transfer data, there is almost always one or more copy operations to move the data between these filesystem pages in kernel space and a memory area in user space. This is because there is not usually a one-to-one alignment between filesystem pages

al type of I/O operation supported by most
ses to take maximum advantage of the page-
ely avoid buffer copies. This is called **memory-**
hings here around memory-mapped files.

establish a virtual memory mapping from user
pages. With a memory-mapped file, you can
that you can access it by simply treating it as a
olifies the code you write in order to modify the

napped files, we start with a RandomAccessFile,
byte buffers are created via the
nds the ByteBuffer class with operations that are
mapped byte buffer and the file mapping that it
lf is garbage-collected. Note that you must
the region that you want to map in the file; this
aller regions of a large file.

```
 Mb

) throws Exception

AccessFile("howtodoinjava.dat", "rw")
hannel().map(FileChannel.MapMode.READ_WRITE, 0,


ing");
```

128 MB long, which is probably larger than the
o be accessible all at once because only portions
rts are swapped out. This way a very large file (up

s can be writable or read-only. The first two
ıd MapMode.READ_WRITE, are fairly obvious.
ıg to be read-only or to allow modification of the
IVATE, indicates that you want a copy-on-write
ıs you make via put( ) will result in a private copy
instance can see. No changes will be made to
will be lost when the buffer is garbage collected.
ents any changes to the underlying file, you must
o a MapMode.PRIVATE mapping. This is
fer object to allow put()s.

od. Once established, a mapping remains in effect

ge collected. Also, mapped buffers are not tied to
associated FileChannel does not destroy the
itself breaks the mapping.

size, but the file it's mapped to is elastic.
ile the mapping is in effect, some or all of the
efined data could be returned, or unchecked
ul about how files are manipulated by other threads
nemory-mapped.

-Mapped Files

es over normal I/O:

s memory, so there is no need to issue read() or

ped memory space, page faults will be generated
from disk. If the user modifies the mapped
automatically marked as dirty and will be
te the file.
operating system will perform intelligent caching
g memory according to system load.
no buffer copying is ever needed.
ut consuming large amounts of memory to copy

ped File

e below code template:

```
"bigFile.xls";
) throws Exception
);
```

```
ode
domAccessFile(file, "r").getChannel();

sing channel.map() operation
nnel.map(FileChannel.MapMode.READ_ONLY, 0, file

as if it were loaded in memory.
d());  //prints false
y());  //Get the size based on content size of

buffer the way you like.
); i++)

r.get()); //Print the content of file
```

Mapped File

ped IO, use below code template:

```
 {
"test.txt";

) throws Exception {

);

a new file


ode
domAccessFile(file, "rw").getChannel();

using channel.map() operation
nnel.map(FileChannel.MapMode.READ_WRITE, 0, 409

hods
tBytes());
```

omments section.

: 2GB. Can I define the offset and split the file
successful doing it. Can you please throw your

42 pm

ng a Memory-Mapped File”? How bog file it was

appedBus
, which supports having multiple processes
y mapped file as well as having multiple

a lot.

ɔytes in large file and I was using a
ıethods and it seems that the MappedByteBuffer

Feedback

equired fields are marked *

java] ... [/java] tags otherwise code may not

Search Tutorials

Search...

Java Input/Output

Password Protected Zip Files

Manage System Log Size

Generate SHA or MD5 Checksum

Deep Cloning using Serialization

Base64 Encode/Decode – Java 8

Java NIO

Java – NIO Introduction

Java NIO vs. Standard IO

Java NIO – Path

Java NIO – Buffers

Java NIO – Channels

Java NIO – Read Files

Java NIO – Streams of Lines

Java NIO – MappedByteBuffer

Java NIO – Vectored IO

Java NIO – Data Transfer between Channels

Recommended

10 Life Lessions

Generate Secure Passwords

How Web Servers work?

How Java I/O Works Internally?

5 Class Design Principles

Unit Testing Best Practices

Exception Handling Best Practices

Best Way to Learn Java

Java Best Practices

Java Interview Questions

Best Java Books

Developer Tools

JSON Formatter and Minifier

XML Formatter and Minifier

CSS Formatter and Minifier

HTML Formatter and Minifier

References

Oracle Java Docs
Spring 3 Reference
Spring 4 References
Jboss RESTEasy JAX-RS
Hibernate Developer Guide
Junit Wiki
Maven FAQs
Dzone Dev Links
JSP Homepage
ANT Homepage

Meta Links

Advertise
Contact Us
Privacy policy
About Me
Online tests