Subject: Re: Sharing and Locking
Posted by mirek on Sun, 21 Mar 2010 13:21:27 GMT
View Forum Message <> Reply to Message

gridem wrote on Sun, 21 March 2010 06:39luzr wrote on Sun, 21 March 2010 09:37
You miss the point: When the file is closed?

(I know when, of course, but the point is the shared ownership makes this very uncertain).
OK, usage sample:

```
void SetterThread()
{
   for (int i = 0; i < cycles; ++ i)
   {
      // create file object
      FileObject file;
      // assign reference to global variable
      *DataAccess::Access() = file;
      // create file itself
      file.Open("file.txt");
      // write some text, file will be opened because accesser doesn't use close
      // (try ... catch is not needed)
      file.Write(String().Cat() << "[" << i << "] setter");
      // close the file, accesser now cannot write into file
      file.Close();
   }
}

void AccesserThread()
{
   for (int i = 0; i < cycles; ++ i)
   {
      try
      {
         // try to get the real object from global reference
         FileObject file = DataAccess::Access()->Get();
         for (int j = 0; j < internalCycles; ++ j)
         {
            // try to write into file
            file.Write(String().Cat() << "[" << i << "," << j << "] accesser");
         }
      }
      catch(Exc& e)
      {
         Out(String().Cat() << "[" << i << "] Accesser error: " << e);
      }
   }
}
```

In the considered implementation the File lifetime is always predictable while lifetime of FileObject can be longer.

See attached file for detailed information.

Regards,
Grigory.

Well, this is the exact tradeoff of GC - you have lost the capability of destructors to manage resources.

Do not get me wrong. What you present is the 'mainstream' approach. In that case, however, the question is why not to use some real GC language instead...

What we are trying to do is exactly oposite. End of block closes the file (pipe, stream, whatever). That is why shared ownership (at interface level) is not recommended...

(P.S.: Not quite sure "try/catch" is not needed there. Who will close the file if the exception leaves the block?)

Mirek
