MonoTorrent - Bug # 449: Issue where the new updated file is smaller than the old one.

Status:	Closed	**Priority:**	Normal
Author:	Yves Bruyere	**Category:**	
Created:	08/18/2010	**Assigned to:**	
Updated:	10/03/2010	**Due date:**	
Subject:	Issue where the new updated file is smaller than the old one.		

Description:

I'm using MonoTorrent in my application to act as a downloader and updater of my Client Files that I have to distribute to my members.

Downloading work most of the time. But I encounter an issue when updating the files.

Here is an exemple to reproduce the issue.

Make first the Test.txt with "Good Day!" to download with MonoTorrent, make an updated file Test.txt with "Hi!" to update with MonoTorrent. The end result of the Test.txt will be "Hi!d Day!". Because the file are not actually shrinked.

To work around this issue, I added some code in the FileStreamBuffer.cs:

@

```
   internal TorrentFileStream GetStream(TorrentFile file, string filePath, FileAccess access)
      {
         TorrentFileStream s = FindStream(filePath);
         if (s != null)
         {
          // If we are requesting write access and the current stream does not have it
            if (((access & FileAccess.Write) == FileAccess.Write) && !s.CanWrite)
            {
               Logger.Log (null, "Didn't have write permission - reopening");
               CloseAndRemove(s);
               s = null;
            }
            else
            {
               // Place the filestream at the end so we know it's been recently used
               list.Remove(s);
               list.Add(s);
            }
         }

         if (s == null)
         {
            if (!File.Exists(filePath))
               SparseFile.CreateSparse(filePath, file.Length);

            s = new TorrentFileStream(filePath, file, FileMode.OpenOrCreate, access, FileShare.Read);
// Code Added
            if (s.Length > file.Length)
            {
               if (!s.CanWrite)
               {
                  s.Close();
                     s = new TorrentFileStream(filePath, file, FileMode.OpenOrCreate, FileAccess.ReadWrite,
```

```
                FileShare.Read);
                    if (s == null) return s;
                }
                s.SetLength(file.Length);
            }
// End
            Add(s);
        }

        return s;
    }
@
```

Maybe it's not the proper way to send patch, but I just wanted to let you know this. Maybe its already patched because I'm using the 0.8 version instead of getting it from the SVN and it's like the SVN do not currently work.

Hope that will help.

History

08/18/2010 03:00 AM - Yves Bruyere

Some code was missing in the last message, here it's where mor exactly I added the code in the function:

```
@    if (s == null)
@        {
@            if (!File.Exists(filePath))
@                SparseFile.CreateSparse(filePath, file.Length);
@
@            s = new TorrentFileStream(filePath, file, FileMode.OpenOrCreate, access, FileShare.Read);
@// Code Added
@            if (s.Length > file.Length)
@            {
@                if (!s.CanWrite)
@                {
@                    s.Close();
@                    s = new TorrentFileStream(filePath, file, FileMode.OpenOrCreate, FileAccess.ReadWrite, FileShare.Read);
@                    if (s == null) return s;
@                }
@                s.SetLength(file.Length);
@            }
@// End
@            Add(s);
@        }
@        return s;
@    }
```

10/03/2010 03:25 PM - Alan McGovern

- Status changed from New to Resolved

Patch committed to git. Thanks! For future reference, the code is now hosted on github and they have a great way to submit patches upstream which I want to encourage people to use. Basically you fork the project on github, make your changes, commit them to your fork and then issue a pull request to me. Then I can review your changes and merge them into the project with a few mouse clicks. There's a few guides on github to explain how it all works.

Thanks again

10/03/2010 03:30 PM - Alan McGovern

- Status changed from Resolved to Closed