MonoTorrent - Bug # 357: TorrentManager.Stop() seems to be asynchronous

Status:	Rejected	**Priority:**	Normal
Author:	Nik Ivanov	**Category:**	
Created:	03/01/2010	**Assigned to:**	
Updated:	03/03/2012	**Due date:**	
Subject:	TorrentManager.Stop() seems to be asynchronous		

Description:

I'm trying to download a file from a multi-file torrent using MonoTorrent. I set the priority of the file to Immediate while all other files are Do Not Download. I monitor the file progress, and once it reaches 100%, I call:

torrentManager.Stop();
torrentEngine.Unregister(torrentManager);

The second line throws an exception:
Exception in mainloop
The manager must be stopped before it can be unregistered
at MonoTorrent.Client.MainLoop.QueueWait(DelegateTask t)
at MonoTorrent.Client.MainLoop.QueueWait(MainLoopTask task)
at MonoTorrent.Client.ClientEngine.Unregister(TorrentManager manager)

Is there a way to synchronously stop a torrent manager, so I can unregister it?

History

03/01/2010 04:41 PM - Nik Ivanov

Forgot to mention:
This happens in MonoTorrent v0.80, release build.

03/03/2010 02:41 PM - Nik Ivanov

The following workaround seems to work at simulating Stop function being synchronous:
This is part of the function where you want to stop and unload the manager:

```
torrentManager.TorrentStateChanged += new EventHandler<TorrentStateChangedEventArgs>(torrentManager_TorrentStateChanged);

torrentManager.Stop();
lock (ManagerUnloadedLock)
{
   Monitor.Wait(ManagerUnloadedLock);
}
```

This is the event handler:

```
private object ManagerUnloadedLock = new object();
void torrentManager_TorrentStateChanged(object sender, TorrentStateChangedEventArgs e)
{
    if (e.NewState == TorrentState.Stopped)
    {
        try
        {
            torrentEngine.Unregister(e.TorrentManager);
            e.TorrentManager.Dispose();
        }
```

```
        catch (Exception ex)
        {
            //handle exception here, shouldnt really happen
        }
        finally
        {
            lock (ManagerUnloadedLock)
            {
                Monitor.Pulse(ManagerUnloadedLock);
            }
}
   }
      }
```

10/03/2010 02:47 PM - Alan McGovern

- Status changed from New to Rejected

It's asynchronous because several long-lasting tasks need to be performed when you stop a torrent. That's why the manager goes from Active -> Stopping and then from Stopping -> Stopped once those tasks have completed. These tasks can take up to 30 seconds to complete. I don't think making 'Stop' synchronous would be a good solution as the common usage of it would be to call it from a UI event and that would make the common usage block the calling application for a long period of time.

02/29/2012 04:02 PM - guyziiz guyziiz

Thanks for the tips, maybe I can use this svelte my noesis marketing and I've been use whatsoever ethnical media in try a interaction and they eff handicraft a big friendliness on me.
Inside Your RV

02/29/2012 04:03 PM - guyziiz guyziiz

Its not the geostationary that commonwealth moldiness be completely fused with communicator's views neighboring musing. So this is what happened with me, anyways its a jazz toil, I revalue it. Thanks
"Inside Your RV":http://www.insideyourrv.com/

03/03/2012 08:59 AM - janooishq janooishq

Its not the geostationary that commonwealth moldiness be completely fused with communicator's views neighboring musing. So this is what happened with me, anyways its a blues toil, I revalue it. Thanks

"BestJobDescriptions.com":http://www.bestjobdescriptions.com/