

Forum:[Hack Forums](#)

Thread: [\[C# & VB.NET\]Understanding Mutex](#)

codepoet - December 14, 2010, 5:25 am

This Tutorial attempts to restrict a Windows application to a single running instance. The article tries to do this using the Process class from the System.Diagnostics namespace. The code invokes Process.GetProcessesByName(processName) to see if there are any existing processes running with the same name, and exits if another is found. If you do some searching, you'll find other code snippets using the same technique.

There are at least three problems with this approach:

- 1) It doesn't account for race conditions. Two instances of the application could launch at nearly the same time, see each other, and both shut down.
- 2) It doesn't work in terminal services, at least not if I want the application to run an instance in each login session.
- 3) It doesn't account for the possibility that someone else might have a process with the same name.

These problems might be considered 'edge conditions', except there is an easy, foolproof way to check for a running instance of an application. A named mutex allows multiple processes to use the same mutex object for interprocess synchronization. The author asserts a mutex is not safe on a multiprocessor machine. If this were true it would be the end of civilization as we know it.

The name of the mutex will preferably be a unique identifier to offset the chances of another application using the same name. One could chose to use the full name of the executing assembly, or a GUID. If the application can acquire the named mutex with WaitOne, then it is the first instance running. If the application calls WaitOne with a timeout value and WaitOne returns false, another instance of the application is running and this one needs to exit.

When using this approach in .NET there is one 'gotcha'. The following code has a small problem:

```
[STAThread]
Code static void Main()
{
    Mutex mutex = new Mutex(false, appGuid);
    if(!mutex.WaitOne(0, false))
    {
        MessageBox.Show("Instance already running");
        return;
    }
    Application.Run(new Form1());
}private static string appGuid = "c0a76b5a-12ab-45c5-b9d9-d693faa6e7b9";
```

The problem is easy to reproduce if you run the following code in a release build:

```
[STAThread]
Code static void Main()
```

```

{
    Mutex mutex = new Mutex(false, appGuid);
    if(!mutex.WaitOne(0, false))
    {
        MessageBox.Show("Instance already running");
        return;
    }
    GC.Collect();
    Application.Run(new Form1());
}

```

```

}private static string appGuid = "c0a76b5a-12ab-45c5-b9d9-d693faa6e7b9";

```

Since the mutex goes unused when the Form starts running, the compiler and garbage collector are free to conspire together to collect the mutex out of existence. After the first garbage collector run, one might be able to launch multiple instances of the application again. The following code will keep the mutex alive. (The call to GC.Collect is still here just for testing).

```

[STAThread]
Code static void Main()
{
    Mutex mutex = new Mutex(false, appGuid);
    if(!mutex.WaitOne(0, false))
    {
        MessageBox.Show("Instance already running");
        return;
    }

```

```

    GC.Collect();
    Application.Run(new Form1());
    GC.KeepAlive(mutex);
}private static string appGuid = "c0a76b5a-12ab-45c5-b9d9-d693faa6e7b9";

```

There is still an imperfection in the code. Mutex derives from WaitHandle, and WaitHandle implements IDisposable. Here is one more example that keeps the mutex alive and properly disposes the mutex when finished.

```

[STAThread]
Code static void Main()
{
    using(Mutex mutex = new Mutex(false, appGuid))
    {
        if(!mutex.WaitOne(0, false))
        {
            MessageBox.Show("Instance already running");
            return;
        }

        GC.Collect();
        Application.Run(new Form1());
    }
}

```

With the above code I can run the application from the console, and also log into the machine with terminal services and run the application in a different session. Terminal services provides a unique namespace for each client session (so does fast user switching on Windows XP). When I create a named mutex, the mutex lives inside the namespace for the session I am running in. Like .NET namespaces, terminal services uses namespaces to prevent naming collisions.

If I want to have only one instance of the application running across all sessions on the machine, I can put the named mutex into the global namespace with the prefix "Global".

```
[STAThread]
Code static void Main()
{
    using(Mutex mutex = new Mutex(false, @"Global" + appGuid))
    {
        if(!mutex.WaitOne(0, false))
        {
            MessageBox.Show("Instance already running");
            return;
        }

        GC.Collect();
        Application.Run(new Form1());
    }
}
```

}After all this you may find you need to adjust permissions on the mutex in order to access the mutex from another process running with different credentials than the first. This requires some PInvoke work and deserves a post unto itself.

Best Regards,
-Crow407

[Credits](#)

codepoet - December 14, 2010, 2:46 pm

50 Views 0 reply... not including my own post.

I'm glad i helped HF Members.

no1me - December 14, 2010, 2:58 pm

try vb net tut, not sure, but real coding post just disappear.

Razorfart - December 14, 2010, 3:04 pm

This is awesome, however I can't say I approve of the amounts of copy-pasteable code in there. But I guess it's necessary.

You clearly know your stuff though, too bad 100's of others that don't will rip this.

affixiate - December 14, 2010, 3:49 pm

(Dec 14, 2010 04:04 PM)Razorfart Wrote: [\[CITE\]](#) This is awesome, however I can't say I approve of the amounts of copy-pasteable code in there. But I guess it's necessary.

You clearly know your stuff though, too bad 100's of others that don't will rip this.

By him knowing his stuff, you mean him "C&Ping" it from:

<http://odetocode.com/blogs/scott/archive/2004/08/20/the-misunderstood-mutex.aspx> , right?

But, it's fine since he gave credits. Just correcting you.

codepoet - December 14, 2010, 4:19 pm

(Dec 14, 2010 04:49 PM)affixiate Wrote: [\[CITE\]](#) By him knowing his stuff, you mean him "C&Ping" it from: <http://odetocode.com/blogs/scott/archive/08/20/the-misunderstood-mutex.aspx> , right?

But, it's fine since he gave credits. Just correcting you.
I KNOW MY STUFF!! THATS WHY I POSTED IT ON <http://odetocode.com/> ?

GET YOUR FACTS STRAIGHT!! ?

tflo - December 14, 2010, 4:45 pm

You posted that six years ago?

posted @ Friday, August 20, 2004 7:22 AM

codepoet - December 14, 2010, 5:14 pm

(Dec 14, 2010 05:45 PM)Vanquish Wrote: [\[CITE\]](#) You posted that six years ago?

posted @ Friday, August 20, 2004 7:22 AM

Spoiler [\(Click to View\)](#)

That would make me 24!!! GOOD JOB. ? NOW YOU KNOW MY AGE!!

Please put a [spoiler] on your image

Exploitations - December 14, 2010, 5:21 pm

(Dec 14, 2010 06:14 PM)Crow407 Wrote: [\[CITE\]](#) That would make me 24!!! GOOD JOB. ? NOW YOU KNOW MY AGE!!

Please put a [spoiler] on your image

Your age on HF says that you're 19.

codepoet - December 14, 2010, 5:31 pm

YES AND MY AGE ON SUPPORTFORUMS.NET IS 26....I HIDE MY AGE FROM FORUMS! IS THAT WRONG?

Try_ME - December 14, 2010, 6:17 pm

I think this is a great tutorial and I love the concept but the problem is that i don't know how to read C# but luckily i converted the code using a neat little website (<http://www.developerfusion.com/tools/convert/csharp-to-vb/>). Thank you

jezebel - December 14, 2010, 6:35 pm

Well you help me out sir Crow407 and this will help me learn C#

codepoet - December 14, 2010, 7:04 pm

(Dec 14, 2010 07:17 PM)Try_ME Wrote: [\[CITE\]](#) I think this is a great tutorial and I love the concept but the problem is that i don't know how to read C# but luckily i converted the code using a neat little website (<http://www.developerfusion.com/tools/convert/csharp-to-vb/>). Thank you

I'm glade i helped..

converters not always good. i prefer to convert it my self.

(Dec 14, 2010 07:35 PM)Parker W Wrote: [\[CITE\]](#) Well you help me out sir Crow407 and this will help me learn C#

What do you need help with?