

# .net 2.0 : looking for a "best practice" for multi threading jobs

---

*Source:*

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.framework/2007-02/msg00633.html>

---

- *From:* "Steve B." <[steve\\_beauge@xxxxxxxxxxxxx](mailto:steve_beauge@xxxxxxxxxxxxx)>
  - *Date:* Fri, 23 Feb 2007 10:38:18 +0100
- 

Hi,

I'm building an application that follow this scenario.

1. Download a file on a server with http
2. Analyse this file and extract other files in one of the section of the file (like dependent files)
3. Foreach found files, repeat the whole process.

The goal is to download the resource and all its dependencies (not know until the first file is downloaded). Each file can have multiple dependencies and one file can be a dependence of several other files.

The process of downloading rely on the `HttpWebResponse` class (I cannot use `WebClient`, because I use some http headers) and is working well.

The process of analysing consists of parsing the file with a custom business object and is also working well.

Everything is working corectly in a "console" application, but I'd like to build onto the process a user friendly form that display progress bar, warning messages, etc.

I also need to reduce the whole process so I want to have several threads working in parralel... in a "pool of object" manner.

The question is : how can I quickly and correctly design my code to have my winform dynamic, and the pool of working threads?

My first reflexion was to store the whole data in a `DataSet` (easy databinding) which can be flat (only on `DataTable` required).

I also created an abstract class "BaseAction" that is inherited by two classes : "DownloadAction" and "AnalyseAction".

My process object defines a `Queue<Action>` where I add all required actions. But I'm confused for the next steps... how can I have multiples trheads that can "Dequeue" actions? moreover, I don't know how to manage the multiples threads since sometimes I'll have only one item in te queue, and this item can add new items...

.net 2.0 : looking for a "best practice" for multi threading jobs

Thanks in advance for any tips.  
Steve