

バッチファイルで使うWAITコマンド的なもの

2021年10月09日

Win10だとWMIの結果が違うっぽいのでソースの

```
while (0 < pcount);
```

を

```
while (1 < pcount);
```

に直して使ってくださいな。

2015年12月26日

.NET Framework 4 がサポートされなくなるので、4.5.2や4.6へのアップグレードを推奨します。公開しているexeは.NET Framework 4.5.2や4.6でも実行可能です。

2014年12月7日

適当にでっち上げた。

GitHubにソースをあげました <https://github.com/k896951/bgwait> 少しずつ慣れてきた。

See [バッチ処理用ツール使い方例](#)

BGWAIT.EXE ダウンロード



BGWAIT.EXEのアーカイブ 2014年12月25日版 まれに自分自身のプロセスがカウントに含まれない場合があるみたい。ループを抜ける条件を変更

BGWAITコマンドの簡単な説明

Windowsのバッチファイルから起動した複数のバックグラウンド実行プログラムの終了まで待ち続けます。UnixのShellにあるwaitコマンドみたいなものです。

sample1.batではa.bat b.bat c.bat をバックグラウンドで起動し d.bat を起動しています。a.bat b.bat c.bat それぞれの実行終了を待たずに d.bat が実行されることになります。もしa.bat b.bat c.bat の終了を待ってから d.bat を実行したい場合、このままでは実現できません。

[sample1.bat](#)

```
@echo off
start a.bat
start b.bat
start c.bat
```

```
call d.bat
```

sample2.batではBGWAITコマンドにより a.bat b.bat c.bat が終了するまで d.bat の実行が行われません。先に作ったPARASTARTコマンドの存在意義を根底から否定するコマンドです。

sample2.bat

```
@echo off
start a.bat
start b.bat
start c.bat
bgwait
call d.bat
```

BGWAITコマンドの実行時条件

bgwait.exe の利用には .NET Framework 4.0 の導入されたWindows環境が必要です。

バッチファイルプログラム(*.bat/*cmd)のほか、コンソールアプリケーションであれば ROBOCOPY のようなプログラムも指定できます*GUIをもつアプリケーションはプロセス終了を検出できません。

オプション指定はありません。

実行時の制限

BGWAITコマンドは、自分の親プロセスと同じ親プロセスIDを持つプロセスをWMIを使って10秒間隔で監視します。バックグラウンド実行させたプログラムが10秒以内に終了していてもBGWAITコマンドの終了まで最大10秒待たされることとなります。

BGWAITコマンドが機能していないように見えるとき

「なんか 機能してないよー」と言う連絡が数回あったので説明。

標準のSTARTコマンドは実行対象がバッチファイルだった場合“CMD.EXE /K バッチファイル”のコマンドラインをバックグラウンドで実行しています。このCMD.EXEの/Kオプションが曲者です。

たとえばa.bat から b.bat をSTARTコマンドで呼び出し、b.bat はWindowsのメモ帳を起動し終了させると文字列“YAHOO!”をコンソールに表示させます。

a.bat

```
@echo off
START /B b.bat
BGWAIT
```

```
PAUSE
```

b.bat

```
@echo off  
notepad  
echo YAHOO!
```

タスクマネージャ等で確認すると判りますが、b.bat は “ CMD.EXE /K b.bat ” のコマンドラインで起動されています。

起動されたメモ帳を終了させると、コンソールに “ YAHOO! ” が表示されますが、そこでなぜかコマンド待ちになってしまいます。a.bat の PAUSE コマンドが実行されておらず、b.bat で終了したように見えます。

```
YAHOO!
```

```
E:\bgwait\bin\Release>
```

実はこれ、b.bat はまだ終了していません。続けて EXIT コマンドを入力するとはっきりします。

```
YAHOO!
```

```
E:\bgwait\bin\Release>EXIT  
続行するには何かキーを押してください...
```

タスクマネージャで確認すると “ CMD.EXE /K b.bat ” の起動プロセス (CMD.EXE のプロセス) が残っています。CMD.EXE の /K オプションは、バッチファイル実行後に CMD.EXE 自身の終了をしませんのでそのまま残っている訳です。

お手軽な対処は、b.bat に EXIT コマンドを追加して CMD.EXE を終了させます。

b.bat

```
@echo off  
notepad  
echo YAHOO!  
EXIT
```

そうすると、メモ帳を終了させたらきちんと a.bat の PAUSE コマンドまで実行されていることがわかります。

```
YAHOO!  
続行するには何かキーを押してください...
```

EXIT コマンドを追加するのは、START コマンドから呼び出すバッチファイルのみにしてください。CALL コマンドで呼び出すバッチファイルの場合、EXIT コマンドがあるとそこで実行終了するので、呼び出し元バッチファイルも実行を終了します。

c.bat

```
@echo off
CALL d.bat
ECHO Google!
PAUSE
```

d.bat

```
@echo off
echo YAHOO!
EXIT
```

c.bat を実行しても“Google!”の表示は行われません。

コンパイル

```
C:\Windows\Microsoft.NET\Framework\v4.0.30319\csc /target:exe
/out:bgwait.exe Program.cs
```

ソース

たいしたことしてない□WindowsのAPI叩くの嫌だったんでWMI使ってる。

GC.Collect()でGCにハッパをかけてメモリ使用を減らすようにしたけどそんなに効いていない。

Program.cs

```
// bgwait
//
// Copyright (C) 2014,2015,2016 Hideki Gotoh ( k896951 )
//
// This software is released under the MIT License.
// http://opensource.org/licenses/mit-license.php
//
using System;
using System.Management;
using System.Diagnostics;
using System.Threading;

namespace bgwait
{
    class Program
    {
        static int Main(string[] args)
```

```
{
    int pcount = 0;
    int ppid = -1;

    try
    {
        string q1 = string.Format("Win32_Process.Handle={0}",
Process.GetCurrentProcess().Id);
        ManagementObject mo = new ManagementObject(q1);
        ppid =
int.Parse(mo.Properties["ParentProcessId"].Value.ToString()); //
parentProcessId
        mo.Dispose();
        mo = null;
    }
    catch (Exception)
    {
        Console.WriteLine("parentProcessId not found.");
    }
    GC.Collect();

    if (-1 == ppid)
    {
        Environment.Exit(16);
    }

    string q2 = string.Format("SELECT * FROM Win32_Process
WHERE ParentProcessId={0}", ppid);
    ManagementObjectSearcher mos = new
ManagementObjectSearcher(q2);
    do
    {
        if (0 != pcount)
        {
            Thread.Sleep(10000); //10sec
        }

        ManagementObjectCollection moc = mos.Get();

        pcount = moc.Count - 1;
        moc.Dispose();
        moc = null;
        GC.Collect();
    }
    while (0 < pcount);

    return 0;
}
}
```

[Windows](#), [batch](#), [tool](#)

From:

<https://wiki.hgotoh.jp/> - 努力したWiki

Permanent link:

<https://wiki.hgotoh.jp/documents/tools/batch/tools-012>

Last update: **2024/11/01 16:25**

