



[General Reading](#) » [Hardware & System](#) » [Windows Services](#) License: [The Code Project Open License \(CPOL\)](#)

VC6, C++Windows, NT4, Win2K, Visual Studio, Dev


# Start Your Windows Programs From An NT Service

By [Xiangyang Liu](#) 刘向阳

Posted: **6 Sep 2000**  
Updated: **22 Feb 2008**  
Views: **2,156,035**  
Bookmarked: **571 times**

Make your MFC, VB and other Windows programs behave like NT services.

639 votes for this Article. 

Popularity: **13.21** Rating: **4.71** out of 5  1 2 3 4 5

 [Download source code - 25.2 Kb](#)

## Introduction

Typically, a Windows service -- used to be called NT service -- is a console application that does not have a message pump. A Windows service can be started without the user having to log into the computer and it won't die after the user logs off. However, it is hard -- sometimes impossible -- to use many existing ActiveX controls within a console application.

On the other hand, MFC and VB applications are Windows applications, so using ActiveX controls in MFC or VB programs is extremely easy. It would be nice to make your MFC and VB programs run like a Windows service, so that:

- They will be started before the user logs onto the computer.
- They will keep running after the user has logged off.

It is possible to write a Windows service as a Windows program, but here I am proposing a much easier solution. I have included with this article the source code for a simple Windows service program that can start and shut down other programs. All you need to do is install this service and modify the INI file. Here are the advantages of using this simple Windows service:

- It can start up to 127 programs of your choice. The started programs behave like Windows services in that they will be running in the background without the user having to log into the machine.
- A user cannot kill the programs started by this service without proper privilege unless, of course, the machine is shutdown.
- You can test and debug your programs outside of the Windows service. For example, you can run your programs in the Visual Studio debugger and step into the source code to find bugs, etc. When it is "bug free," you can deploy it in production, starting it from the Windows service.
- You can run a broad range of programs using this service, including .NET programs, VB scripts and batch files. This service has been used by the open source world to run numerous Java programs.

## A little story

My desktop PC in the office was recently re-imaged by the support folks of my company. The first time I signed in, I noticed that a Java program was busy doing something in the background. When I looked deeper, it became clear that this Java program was started by a modified version of the Windows service explained in this article. Later, I found out that another development team in my company got this "great open source tool" from the internet and they modified it to run their Java programs :-).

## XYNTService

*XYNTService.exe* is the name of the executable for this Windows service program. You may get this Windows service from other websites, but Code Project is the official place to get the most recent version. You can freely use and modify the source code included with this article. I am aware that there are other utility programs providing almost the same functionality as XYNTService. However, as you will see, XYNTService has more features and is a lot easier to use. For example, no editing of the registry is required. Here is how to use the program:

- To install the service, run the following at the command prompt: `XYNTService -i`.
- To un-install the service, run the following at the command prompt: `XYNTService -u`.

By default, the installed service will be started automatically when you reboot the computer. You can also start and shut down the service from the Control Panel or the Administrative Tools using the Services option. When the service is started, it will create all of the processes you defined in the *XYNTService.ini* file one by one. When the service is shut down, it will terminate each of the processes it created in reverse order. The *XYNTService.ini* file should be placed in the same directory as the executable. Here is a sample of the file:

```
[Settings]
ServiceName = XYNTService
CheckProcessSeconds = 30
[Process0]
CommandLine = c:\winnt\system32\notepad.exe
WorkingDir= c:\
PauseStart= 1000
PauseEnd= 1000
UserInterface = Yes
Restart = Yes
[Process1]
CommandLine = java.exe MyPackage.MyClass
Restart = No
UserName =
Domain =
Password =
```

The `ServiceName` property specifies the name you want to use for this NT service. The default name is `XYNTService`. If you copy the executable and the INI file into a different directory and then modify the `ServiceName` property in the INI file, you can install and configure a different service!

The sections `[Process0]`, `[Process1]`, ..., etc. define the properties related to each of these processes. As you can see, there are two processes to create in this example. *notepad.exe* and *java.exe* are the names of these programs. You can specify the parameters for each of these processes in the `CommandLine` property. You **must** specify the full path of the executable file for the corresponding process in the `CommandLine` property unless the executable is already in the system path.

The `CheckProcessSeconds` property specifies if and how often you want to check the processes started by XYNTService. If the property has value 0, then no checking is done. If, for example, the property value is 30 then every 30 seconds XYNTService will query the operating system to see if the processes it started are still running. The dead ones will be restarted if the `Restart` property value (explained later) is defined as `Yes` for that process. The default value of this property, if you don't specify it, is 600 (10 minutes).

**Note:** In the previous versions of XYNTService, the `ProcCount` value specified how many processes were started by XYNTService. This is no longer required. The `CheckProcess` value is the number of minutes instead of seconds. The current version will work as before if you use `CheckProcess` instead of `CheckProcessSeconds`; you should not use both.

The `WorkingDir` property is the working directory of the current process. If you don't specify this property, then the working directory of the current process will be `c:\winnt\system32`. The `PauseStart` property is the number of milliseconds the service will wait after starting the current process and before starting the next process. This is useful where the next process depends on the previous process. For example, the second process has to "connect" to the first process in such a way that it does not run until the first process is finished with initialization. If you don't specify the

`PauseStart` property, the default value will be 100 milliseconds.

When XYNTService is shut down, it will post `WM_QUIT` messages to the processes it created first and then call the Win32 function `TerminateProcess`. The `PauseEnd` property is the number of milliseconds the service will wait before `TerminateProcess` is called. This property can be used to give a process (started by XYNTService) a chance to clean up and shutdown itself. If you don't specify the `PauseEnd` property, the default value will be 100 milliseconds.

The `UserInterface` property controls whether a logged-in user can see the processes created by XYNTService. However, this only works when XYNTService is running under the local system account, which is the default. In this case, processes created by XYNTService will not be able to access a specific user's settings, such as mapped network drives, etc. You can configure XYNTService to run under a user account, which is done easily from the Control Panel or Administrative Tools. Just select Services and then double click XYNTService in the installed services list to bring up a dialog box.

The `Restart` property is used to decide whether you want XYNTService to restart a dead process. If this property is `No`, which is the default if you don't specify any value, then the corresponding process will not be restarted. If this property is `Yes`, then the dead process will be restarted by XYNTService. See the `CheckProcess` property above on how often dead processes are restarted. You can **bounce** (stop and restart) any process defined in the INI file from the command line. For example, the following command...

```
XYNTService -b 2
```

...will stop and restart the process defined in the `[Process2]` section of the INI file. XYNTService can also be used to start and stop other services from the command line. Here are the commands to start (run) and stop (kill) other services:

```
XYNTService -r NameOfServiceToRun
XYNTService -k NameOfServiceToKill
```

In particular, you can use the above commands to start and stop XYNTService from the command line! **Important note:** You cannot start XYNTService by running it from the command prompt without an argument.

All errors while running XYNTService are written into a log file in the same directory as the executable. The error code in the log file is a decimal number returned by the `GetLastError` API. You can look for it on [MSDN](#).

## Impersonating a user (new feature)

The `UserName`, `Domain`, and `Password` properties are used to impersonate a user. If `UserName` and `Password` are not empty, then the corresponding process will be started using the specified user account. If `Domain` is empty, then the specified user account must be an account on the local machine instead of a network account. In order to impersonate a user, the service must be run by a local system. Also, the corresponding user account must have the "logon as service" privilege. If you configure a service to run under a user account, that account will acquire the "logon as service" privilege. Please note that you won't be able to see the user interface of an impersonated process.

Since we may need to put a user password in the INI file, it is important to make sure that the INI file is only accessible by administrators and the local system. The following command will protect the current folder, making it accessible only by administrators and the local system.

```
cacls . /g system:F administrators:F
```

## Frequently asked questions

- **Why can't XYNTService start my program?** There could be many reasons. The likely most ones are that you did not give the correct path of the executable in the `XYNTService.ini` file or

that your program is located on a mapped network drive (see the following question).

- **My program works fine outside of XYNTService; why does it fail when started by XYNTService?** XYNTService is running under the "local system" account by default. Any program started by it will also use this account. Your program may need some resources that are not available to this account. For example, "local system" cannot access the current user's registry settings, nor can it access a mapped network drive or any other resource on the LAN. However, you can change the account used by XYNTService. This topic is covered in the question below.
- **How do I change the account that XYNTService uses?** On Windows 2000 or XP, use the Administrative Tools menu, select Services and then double click XYNTService from the displayed list to change the logon information (domain name, user name and password). On Windows NT 4.0, use the Control Panel, select/open the Services icon and then double click XYNTService from the displayed list to change the logon information. Please note that you cannot see your program started by XYNTService if XYNTService is not using the "local system" account.
- **How do I turn my program into a Windows service?** You can't, unless you rewrite your program from scratch. A service is a special program that requires some special knowledge to write. If you don't want to learn how to write a service, then use XYNTService to start your program as described in this article so that your program behaves like a service.
- **How do I debug a service?** First, build your service program in debug mode from Visual Studio and set break points at the appropriate places. Then install and start the service. Attach the debugger to the service executable, which should already be running. Note that a service has to be started before the process can be attached to the debugger, so some parts of it can never be stepped into from the debugger. In fact, the most useful code in XYNTService cannot be debugged because the code is executed while XYNTService is started.
- **Why won't my Java program run under XYNTService or why does it die when the user logs off the machine?** I think this could be due to a bug in Java itself. I had some problems with recent versions of JDK, but not with JDK 1.2.2. If you cannot use JDK 1.2.2, then try the *Xrs* option when using Java. Also, change the account used by XYNTService to a LAN user instead of the default "local system" account. This may help you to get around the problem.

## History

- 22<sup>nd</sup> Feb, 08
  - Download recompiled in VS2005 to stop bogus anti-virus warnings.
- 6<sup>th</sup> Jun, 07
  - Updated source files and article.
- 12<sup>th</sup> Jan, 07
  - Updated article.
- 24<sup>th</sup> Feb, 06
  - Updated source code and article.
- 28<sup>th</sup> Feb, 04
  - Updated source files.
- 7<sup>th</sup> Sep, 00
  - Original version posted.

## License

This article, along with any associated source code and files, is licensed under [The Code Project Open License \(CPOL\)](#)

## About the Author

**Xiangyang Liu** 刘向阳

Location:  United States

Msgs 226 to 250 of 1,307 (Total in Forum: 1,307) ([Refresh](#))[First](#) [Prev](#) [Next](#)[Re: Great work](#)[Power Puff Boy](#)

4:24 27 Feb '07

Dr. Luiji said what needs to be said.  
I didn't feel like dealing with "how to write an NT service"... It helped me out on that one!

A thousand thanks to the author.

[Re: Great work](#)[gfilkov52](#)

12:43 16 Mar '07

Agree!

I had to modify it slightly, though.  
The server application I need to install as service is a Java one.  
It takes separate batch files to start the server and to stop it.  
After my modification it works fine! Just like a native EXE.  
I can start and stop the server from system Services management console, as well as from the programm shortcuts.

Thanks.

Guennadi

[Re: Great work](#)[Xiangyang Liu](#)

18:44 19 Mar '07

Thank you, Dr. Luiji.

---

[My .NET Business Application Framework](#)  
[My Home Page](#)

**Service starts but do not detect the application has been stopped**[jmivielle](#)

9:35 13 Feb '07

Hello,

I installed XYNTService on a Windows 2003 server, and the process I want to be start as a service is an API Server, launched by a java command line. When I start the matching XYNTservice (from the Windows admin tools), it works fine, and I can acces to the web administration of the server.

But when I stop the server (manually, ie from the web administration), the service does not detect that and so does not restart. On the same way, when I stop the service (from the windows admin tool), the API server is still running.

The java command line is something like:

```
java -cp classpathfiles  
com.atosorigin.services.cad.apiserver.watchdog.ProcessWatchdog
```

I think the problem is that I do not launch a basic application or a standard java application, but a server ...  
Have you any idea to solve it ? Cause I really need to launch this server as a service ...

Thanks by advance ! 😊

Jérôme

 [Re: Service starts but do not detect the application has been stopped](#)  [IshtiaqHossain](#) 23:26 14 May '07

I think I'm having the same problem. I have a program written in Java that basically opens a port and starts listening. I converted the jar file to .exe and now wish to run it as a service. After using XYNTService, the .exe started running as a service but the problem occurred when i tried to stop it from the windows admin tool. It shows that the service has stopped but it actually continues to run. I have to kill javaw.exe from the task manager to make it stop. Is there something wrong with my code ? how can i automaticcaly kill javaw.exe for this process when it stops? need an urgent solution.

Ishtiaq

 **Failed to work with Vista.**  [chueh8](#) 21:23 28 Jan '07

openSCManager failed, error code=0x05

 [Re: Failed to work with Vista.](#)  [Xiangyang Liu](#) 17:59 11 Feb '07



What is Vista? 😊

[My .NET Business Application Framework](#)  
[My Home Page](#)

 [Re: Failed to work with Vista.](#)  [NelsonKostaSouto](#) 1:21 12 Feb '07

Vista is the New Windows from Microsoft.  
In this case you need to upgrade the code to Visual Studio 2005 😊

NKS

 [Re: Failed to work with Vista.](#)  [djsdjs](#) 15:02 16 Mar '07

I am getting the same error under Windows Vista. It looks like the person who added this comment upgraded the source so the compiled version would work under vista - any chance of getting that posted?

Thanks,  
D.

 **Updated to Visual Studio 2005**  [NKS2007](#) 5:04 17 Jan '07

Hi,



I updated the source code to visual studio 2005.  
How can i send you the source code?

NKS

 [Re: Updated to Visual Studio 2005](#)  [Xiangyang Liu](#) 18:04 11 Feb '07

I don't see any advantage of migrating the code to VS 2005. If you rewrite it or added new features, you can post the code in your own article.

[My .NET Business Application Framework](#)  
[My Home Page](#)


 [Re: Updated to Visual Studio 2005](#)  [charlot](#) 1:53 23 Mar '07

Hello,  
I'm an italian developer. I'm interested in XYNTService and I've read you have update the source code to visual studio 2005.  
It's possible you can share with me your conversion (I've to do this thing too for an expansion of the program).

Thank you for any answer.

Best regards

charlot@mailcity.com

 **Updated to Visual Studio 2005**  **5:02 17 Jan '07**

Hi,

I upgrade the source code to Visual Studio 2005.

How can i send the source code to you?

NKS

 **VS2005 C++ Code Snippets availability.**  **oleg63** **8:42 16 Jan '07**

Hi,

I want to use the Code Snippets in Visual C++ VS2005 Professional.

But it seems it's not exist here 😞

When I go menu Tools\Code Snippets Manager I can not select a C++ from the "Language" combo box.

Is there some way to activate the Code Snippets stuff in C++, or it's available in another editions of Visual Studio 2005(for example VS2005 Developer Edition)?

Thanks.

 **Great service but what about srvany ?**  **ChrisRibe** **8:07 16 Jan '07**

Hi again great service !

But what about srvany.exe?

<http://support.microsoft.com/kb/152460>

Does it not do the same thing as XYNTService? Mind you srvany.exe does not seem to have all the options that XYNTService has but I had to ask...

Kinda looked dumb when the IT guy told me about this 😊

Thanks for the reply

Chris 😊

 **Re: Great service but what about srvany ?**  **Xiangyang Liu** **10:46 17 Jan '07**

**ChrisRibe wrote:**

But what about srvany.exe?

I wrote this tool six and half years ago. I did not know anything about SrvAny at that time. I heard about it from a reader and went back to work, adding more features I believed not present in SvrAny.

I still don't (need to) know much about SvrAny. 😊

Thanks.

[My .NET Business Application Framework](#)  
[My Home Page](#)



 [Re: Great service but what about srvany ?](#) [ChrisRibe](#)

6:02 18 Jan '07


Thanks for the reply,

I like this service better than srvany anyhow.

Now that you mention it you posted this \*six and a half\* years ago ! Microsoft probably got wind of this service/need and \*tried\* to fill the void.

Thanks for your dedication to this article. 😊

Chris

 [Re: Great service but what about srvany ?](#) [AustinFlyer](#)

8:16 19 Jan '07

I'm not a Windows expert by any stretch of the imagination, but I initially tried srvany, and it wouldn't fully satisfy my requirements. At least I couldn't figure out how to make it satisfy them (my real strength is UNIX). After some Googling I found this project, and it worked great for me.

Briefly, my requirements were to take a current process where a master batch file is invoked manually to start five other background processes in order, and turn it into something that starts at system boot and runs as a specific user.

The main problem I had with srvany is that stopping the service does not stop the background processes. I thought about starting each of the five background processes as a separate service, but wasn't sure how to define the dependencies. XYNTService's .ini file was very easy to use, and it correctly shut down my other processes when I stopped the service.

Thank you for posting this service, Xiangyang Liu.

 [Re: Great service but what about srvany ?](#) [ChrisRibe](#)

7:11 23 Jan '07

Thanks AustinFlyer for posting your user experience with srvany and why you preferred XYNTService's.

Saves me the time and frustration of testing out srvany 😊

Chris

 **Update 1/9/2007** [Xiangyang Liu](#)**12:47 9 Jan '07**

Updated version of download zip file (source code and exe) has been submitted to the webmaster. This update fixes the problem of unclosed handles for the child processes.

Thanks.

[My .NET Business Application Framework](#)  
[My Home Page](#)

 **Leak and Error Code 1450 Solved [modified]** [JorgeCordero](#)**0:46 9 Jan '07**

Hi Xiangyang,

Congratulations, excellent work. But there is still an important problem: unnecessary handles not properly closed. So far, Chris Amor, on 10:21 19 Jun '06 post, has the best solution.

Please include it in the code.

Remember, the 1450 error is ERROR\_NO\_SYSTEM\_RESOURCES (in our case, unnecessary open handles, exhausting the system resources).

As the app allows it, many people may be using it as a task bouncer, constantly calling again and again a short application. In this particular case, eventually resources will go off.

Sebastian

-- modified at 3:52 Tuesday 9th January, 2007



Sebastian

 [Re: Leak and Error Code 1450 Solved](#)

 [Xiangyang Liu](#)

12:18 9 Jan '07

Thank you very much!

I will post the updated source to the webmaster soon.

P.S. I read the other people's messages but was distracted and left the problem unfixed for a long time. Sorry.

[My .NET Business Application Framework](#)  
[My Home Page](#)

 **XYNTServiceWrapper starts great but not XYNTService**

 [adelara](#)

**11:31 29 Dec '06**

Am I missing any switch to start my configuration ?  
 I actually am testing the example with notepad that can be found in c:\windows\notepad.exe. Of course, it works from the command line. Now, starting XYNTService with the .ini settings it just sits there and notepad never starts. Then I found the wrapper and used it to review the configurations that were all ok. Finally I hit the 'Start' button on the GUI and things work the way they are supposed to: notepad starts and if I kill, XYNTService will restart it again.

The bottom line: when I use "XYNTService -i" from the command line, it installs but 'notepad' never comes up, therefore I was wondering if there is another switch to be added to the cli invocation. Thanks,  
 -Alex

 [Re: XYNTServiceWrapper starts great but not XYNTService](#)

 [JorgeCordero](#)

1:23 9 Jan '07

XYNTService -i will just install the app as a service. In other words, it would register a new entry in the service database of your server. Use windows to start the service, or just reboot, as a proper test.

Also XYNTService -r would be fine, but if you need to test the production scenario, use the above criteria.

Good coding (or testing)

Sebastian

 **How to start a VPN connexion in XP ?**

 [mpconti](#)

**7:22 21 Dec '06**

I'm using XYNTService and I need to start VPN connexion to start application that transfer files automatically to server for processing.  
 How I configure ini file ?

Marcelo Pires Conti

Last Visit: 16:59 16 Sep '08 Last Update: 0:04 9 Feb '09

[« Prev](#) [1](#) ... [6](#) [7](#) [8](#) [9](#) **[10](#)** [11](#) [12](#) [13](#) [14](#) [Next »](#)

 General  News  Question  Answer  Joke  Rant  Admin

[PermaLink](#) | [Privacy](#) | [Terms of Use](#)  
Last Updated: 22 Feb 2008  
Editor: [Chris Maunder](#)

Copyright 2000 by Xiangyang Liu 刘向阳  
Everything else Copyright © [CodeProject](#), 1999-2009  
Web16 | [Advertise on the Code Project](#)