

ABC-Deploy

ABC_SrvAny.Exe Guide

Simple as your first ABC
Flexible as an alphabet

2/1/2010
ABC-Deploy.Com

ABC-Deploy ABC_SrvAny.Exe Guide

Table of Contents

ABC_SrvAny.Exe	3
Description:	3
Requirements	3
Installation:.....	3
Naming the service	4
Specifying the application to start & its parameters:	4
Starting & stopping the service:	4
UnInstall the Service:	5
Programming Considerations:.....	5
Comments/limitations:	5
Legal Stuff.....	7

ABC_SrvAny.Exe

Description:

This document describes the free ABC_SrvAny.Exe utility.

This utility allows running Windows applications as services. The benefits include:

- allow applications to survive logoff/logon sequences, hence saving the overhead of re-starting them for each new user
- allow server applications to come-up and service requests even when no user is logged-on
- allow applications to run and perform a task in a specific logon account, different from the currently logged-on user

ABC_SrvAny.Exe are being delivered to you as part of the desktop management system "ABC-Deploy.Com" "ABC-Deploy.Com" enables you as an network administrator to easily automate management of your networked Windows Clients,

GoTo <http://abc-deploy.com> for further details.

Requirements

ABC_SrvAny.Exe will run on any Windows version since Windows NT 4.0

ABC_SrvAny.Exe installs on both 32 and 64 bits Windows, and are able to start 32 bit as well as 64 bit applications as supported by the operating system

ABC_SrvAny.Exe are written solely using Windows native API's no .NET FrameWork or VMI are required on the client

Installation:

- ➔ Copy ABC_SrvAny.Exe to your system and install it as a Windows Service, for example:
 - `C:\tools\ABC_SrvAny.Exe /Install`
 - `C:\tools\ABC_SrvAny.Exe /Install /Silent` [For silent installation]
- ➔ Configure via the Services applet ("Startup..." dialog) of the Control Panel as disabled, manual or automatic, as appropriate.
- ➔ Set via the Services applet ("Startup..." dialog) of the Control Panel the account for the service. If you need access to the screen & keyboard, you must choose the LocalSystem account and click the "Allow Service to Interact with Desktop", otherwise choose any account and specify the correct logon password. Note that the LocalSystem account might have limited network access.

Naming the service

A Windows service has two names.

The long name you see in the Control Panel is the display name of the service. The internal shorter name of the service is often called the key name.

When you install your service using *ABC_SrvAny.Exe /Install* your newly created service will be named "ABC_SrvAny" with a DisplayName set as "ABC-SrvAny"

To use a service name of your choice, you must rename the ABC_SrvAny.Exe file to have the name of your service. Rename to ie. MyService.Exe – Then your service will get the key name "MyService"

Use the Registry editor to set a nicer looking DisplayName if you want. Change this value:

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\MyService\DisplayName

Specifying the application to start & its parameters:

1. By Editing the registry
 - ➔ Locate the following key
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\MyService\Parameters
 - ➔ Under the above key, you will find an 'Application' value of type REG_SZ that specifies the full path of your app executable (including the extension). For example:
Application: %WinDir%\NotePad.Exe
 - ➔ OPTIONAL: under the above key, you will find an 'Application Parameters' value of type REG_SZ that specifies the parameters for your app. For Example:
Application Parameters: %Temp%\ ABC_SrvAny.Txt
 - ➔ OPTIONAL: under the above key, you will find an 'Application Default' value of type REG_SZ that specifies the current directory to use for the app. For Example:
Application Default: %HomePath%

Starting & stopping the service:

Start:

If the service is configured as 'Automatic' (The default), the user doesn't need to start it explicitly: it is started automatically every time when the system is re-booted.

For 'manual' services, the user may start services in several ways:

- from the Services applet of the Control Panel
- or
- using the SC.Exe utility, eg: *SC start ABC_SrvAny*
- or
- *NET START ABC_SrvAny*

Stop:

When you stop the service, it will terminate the application it had started. The way to stop the service is:

- from the Services applet of the Control Panel
- or
- using the SC.Exe utility, eg: *SC stop ABC_SrvAny*
- or
- *NET STOP ABC_SrvAny*

WARNING:

When the service is stopped, it terminates the application via the Windows TerminateProcess API. This is a drastic way to end an application. For example, it would not allow the application to prompt the user to save changes. Therefore, it is recommended to close the application BEFORE stopping the service.

UnInstall the Service:

If you want to prevent an instance of the ABC_SrvAny utility from running until further notice, you should configure it via the Services applet ("Startup..." dialog) of the Control Panel as 'Disabled' (rather than 'manual' or 'automatic').

If you want to remove an instance of the ABC_SrvAny utility permanently:

- ➔ If the service is running, stop it (see "Starting and stopping the service" above)
- ➔ Run:
 - `C:\tools\ABC_SrvAny.Exe /UnInstall`
 - `C:\tools\ABC_SrvAny.Exe /UnInstall /Silent` *[For silent UnInstall]*

NOTICE: this procedure removes only one specific instance of ABC_SrvAny. Therefore, it is possible that ABC_SrvAny.Exe is still being used to start other applications as services.

Programming Considerations:

Some applications may terminate upon logoff, even though they were started as a service, if they don't ignore the WM_ENDSESSION message (or CTRL_LOGOFF_EVENT).

For WIN32 graphical applications: when the currently logged-in user is logging-off, all WIN32 top-level windows receive WM_QUERYENDSESSION and WM_ENDSESSION messages.

Some WIN32 applications choose to terminate upon receipt of such messages. In order for your WIN32 application to survive logoff, it must not do that: instead, your windows procedure should call the default windows procedure on these messages.

For WIN32 Console (i.e. character-mode) applications: when the currently logged-in user is logging-off, all Console applications receive a CTRL_LOGOFF_EVENT event from the Console.

If your Console application has registered a Console event handler (via SetConsoleCtrlHandler), it must ignore CTRL_LOGOFF_EVENT in order to survive the logoff.

Comments/limitations:

- ➔ You may install ABC_SrvAny.Exe several times with different registry parameters (i.e. running different target applications)
To use a distinct service name for each instance (e.g. MyService1, MyService2 etc.) you must copy the ABC_SrvAny.Exe file to a new instance with the name of your service ie.: MyService.Exe
Afterwards you will use *MyService.Exe /Install* and *MyService.Exe /UnInstall* to add and remove the service
- ➔ If ABC_SrvAny.Exe fails to start your application, try specifying as current directory the directory where your application is installed (see 'AppDirectory' registry key) ABC_SrvAny.Exe may be running under an account different than the currently logged-on user therefore environment variables may

be set differently: as a result, for example, the system might have been unable to find a DLL required for your application and running it from the application's directory might help.

- ➔ When you stop the service, it will terminate the application it had started
- ➔ Due to security restrictions in Windows Vista and in Windows 7, Services GUI elements (Such as Windows) will not be directly visible to the logged on user.

On NT, Win2K, XP and Win2K3 the console user is always logged on in Session 0, the same session the services live in. If a ABC_SrvAny service is configured as interactive, it'll be able to show the GUI direct on the user's desktop. This however is not possible in Windows Vista or in Windows 7

Users of Windows versions after with Vista will however be able to see the GUI by clicking a "Interactive Services Detection" icon that displays in the taskbar. Hereby enabling access to dialogs created by interactive services when they appear.

The Windows "Interactive Service Detection" service is required to be running to allow interactive service dialogs. If this service is disabled, both notifications of and access to new interactive service dialogs will no longer function.

- ➔ Due to a restriction enforced by Windows on services, the application can either be interactive (have a Console, read keyboard input etc.) or have network access (not both at the same time).
- ➔ This limitation is less problematic than it would appear at first glance. Here is why:
 - If you know which servers your interactive service application needs to access, it is possible to configure these servers (or selected shares on them) to allow your service (running under the LocalSystem account) to access these servers.
 - a. To allow LocalSystem services on any machine in the domain to access a specific share on a server, use the Registry Editor to add the name of that share to:
 - HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\LanmanServer\Parameters\NullSessionShares
 - If named pipes on that server also need to be accessed by LocalSystem services, add them to:
 - HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\LanmanServer\Parameters\NullSessionPipes
 - b. To allow ALL shares & pipes on the server to be accessed by LocalSystem services, add a value:
 - HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\LanmanServer\Parameters\RestrictNullSessAccess, type DWORD, set to value 0
 - WARNING: method b) effectively allows everyone in the domain to access that server - make sure this is acceptable for you.

Legal Stuff

ABC_SrvAny.Exe is 100% free to use for both commercial and none commercial use.

You may redistribute ABC-SrvAny.Exe in one of two ways:

- Copy this document, together with the ABC_SrvAny.EXE program
- Point users to our download location at <http://ABC-Deploy.Com>