# 4. Managing application pools

Let’s face it, as an administrator your biggest concern with IIS is this: will it keep running your sites and applications reliably? Have you ever had a misbehaving application, one the leaks memory and starts to slow down your computer? Without dealing with the situation, your computer will eventually crash and need to be rebooted. Websites and applications are no different. Some of them are well-behaved little children running on your web server. Others are nightmares eating up memory and hogging processing. Without the ability to separate the good from the bad, you’d find yourself going to the office in the middle of the night to restart your web servers.

*Application pools* provide isolation to each website on a server, preventing one site from harming (crashing) another. Using them increases the web server’s reliability and the availability of each website. Think of the virtual machines you’re using for the labs for this book; each one has its own memory and processing allocation. If one VM crashes, it has no effect on the others, nor does it crash the host operating system.

You can use application pools to isolate websites and applications in several scenarios, such as the following:

* Isolating well-behaved applications from unstable ones
* Increasing security by preventing one application from accessing the resource of another
* Increasing security by assigning unique identities to pools
* Grouping websites and applications that have the same pool configuration settings

In this chapter I show you how to work through such scenarios, using the bicycle shop website your started building in chapter 3. You’ll focus on creating and configuring new application pools, setting the best security for the pools, and managing the recycling and cleaning of the pools.

So open your lunch sack and let’s get started with creating and configuring application pools.

## 4.1 Creating and configuring standard application pool settings

Application pools have basic configuration settings that will work for most websites and applications right out of the box. Some applications – for example, misbehaving ones that crash often – need to have their settings tweaked to increase their reliability. I show you what to look for after examining the default settings. This section focuses on the basic application pools settings, understanding which ones to use and alter for the websites, and applications that comprise the environment of the bike shop.

### 4.1.1 Locating application pools and settings

When IIS was installed, it created the default website we explored and began adding the bike shop to in the last lunch. Fortunately it also created a default application pool we can explore. In this section I show you where to find the application pools used for your websites and web applications, both in the graphical manager and using PowerShell.

**FINDING APPLICATION POOLS IN THE GUI TO MANAGE SETTING AND CONFIGURATION**

In figure 4.1 you can see Application Pools under the navigation pane. When you select this, the center pane lists all the currently defined application pools and the basic settings for them. You’ll create new application pools for new websites and new web applications for the bike shop here. You’ll view both basic and advanced settings in the action pane.

**FINDING APPLICTION POOLS USING POWERSHELL TO MANAGE SETTINGS AND CONFIGURATION**

In PowerShell using the WebAdministration module, you can gather the same information. Much of the IIS information is stored under a PowerShell drive called IIS:. You can access that drive and its information by navigating the directory system or directly using the Get-Item cmdlet.

To navigate the IIS: PowerShell drive, do this:

PS> Set-Location IIS:\appPools

PS> Get-ChildItem



Figure 4.1 Locating application pools and the configuration settings

TRY IT NOW Put down your sandwich for a minute and open the IIS manager. Navigate to the default application pool and examine the basic settings by viewing them in the center pane or selecting Basic Settings in the actions pane. While you’re at it, open up a PowerShell console, import the WebAdministration module, and locate the default application pool using filesystem.

Now that you’ve located the default application pool and its basic settings, let’s decipher what those settings do. You’ll understand those settings better if we start by creating a fresh, new application pool.

### 4.1.2 Creating a new application pool

Placing websites, and sometimes applications, in their own application pools is the most reliable and most secure option. Application pools are isolated from each other to prevent one application from bringing down another. As an added security benefit, application pools also prevent one application from sneaking into another pool and stealing information. The best practice is to place new websites into their own application pools.

**CREATING AN APPLICATION POOL WITH THE GUI**

You don’t have multiple websites ye for WebBikez (you’ll do that in chapter 5), so rather than use the default application pool, let’s experiment with one of our own using both the GUI and PowerShell. You create a new application pool in IIS manager by selecting Add Application Pool from the actions pane. A menu pops up prompting for a new name for the application pool, plus some addition settings, as shown in figure 4.2.



Figure 4.2 Basic application pool settings

Let’s look at each of these details, starting with the name.

**NAME**

The application pool name can be any unique alphanumeric name of your choice. The best practice is to name your application pool the same as the website or application that will run inside the pool, adding the suffix *pool*. That makes it easy to keep track of which pool goes with which website or application.

As an example, in chapter 5 you’ll be creating several new websites and applications. One of the websites might be named WebBikezBags. A great name for the application pool would be WebBkezBagsPool. I prefer not to add spaces, dashes, or underscores to an application pool name. Although it may be easier to read visually, it’s harder to deal with when using PowerShell or other command-line utilities.

In figure 4.2 I named the pool MyAppPool because I’m not using it for an application yet. You can always rename an application pool later.

**.NET FRAMWORK VERSION**

Many of the applications and websites that you’ll host on IIS require a special library called the .NET Framework. Developers use this library to provide functionality for their applications. The application pool will load the library so that the applications will work properly.

|  |
| --- |
| **Above and beyond**  An application pool can only support one version of the .NET Framework. All applications that run inside the pool must support the same version. If you need to have two applications that need two different framework versions, create a new web application and application pool for each. |

You’ll need to select the correct version of the .NET Framework for your application. How do you know what is the correct version? You need to check the application installation documentation to know for sure. For this bike shop, the developers are developing the sites using .NET 4.0. Not all applications require the .NET Framework – in fact you may have several that don’t. You’ll explore those types of applications in later chapters.

Here are the options you’ll see when you click the .NET Framework dropdown:

* ***No Managed Code*** – Select this if your application or website doesn’t require .NET Framework support.
* ***V.20*** – Use this version of the framework to support applications written with .NET Framework versions 2.0, 3.0, or 3.5.