Configure Web gardens in IIS 7 step by step

By Md. Mahedee Hasan, Senior Software Engineer, LEADS

Introduction

“Web garden” scaling out by increasing the number of worker processes per CPU. Web gardening will allow more than two subscribers to synchronize at the same time. Before going to the details of web garden, I hope you have fundamental idea on application pool and worker process. Just for reminder - for request processing in IIS, worker process (w3wp.exe) runs the ASP.NET application in IIS. Worker process is responsible for handling all kinds of request, response, session data, and cache data. Application Pool is the container of worker process. Application pool is used to separate sets of IIS worker processes and enables a better security, reliability, and availability for any web application. It will also increase CPU utilization by replisapi.dll, which can negatively impact overall server performance. So, it is important to balance these considerations when you choose a value for Maximum Worker Processes.

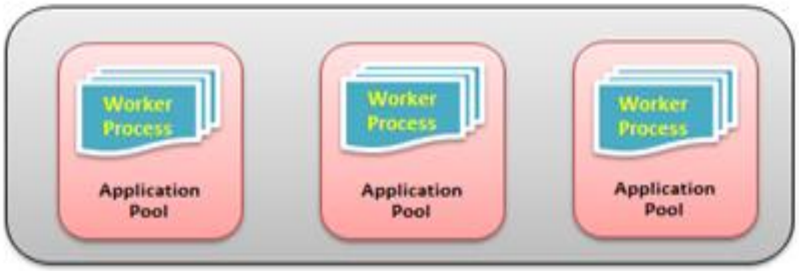


Fig – Application pool is the container of one or more worker processes

Now, by default, each and every Application pool contains a single worker process. Application which contains the multiple worker process is called **“Web Garden”**. Below is the typical diagram for a web garden application.

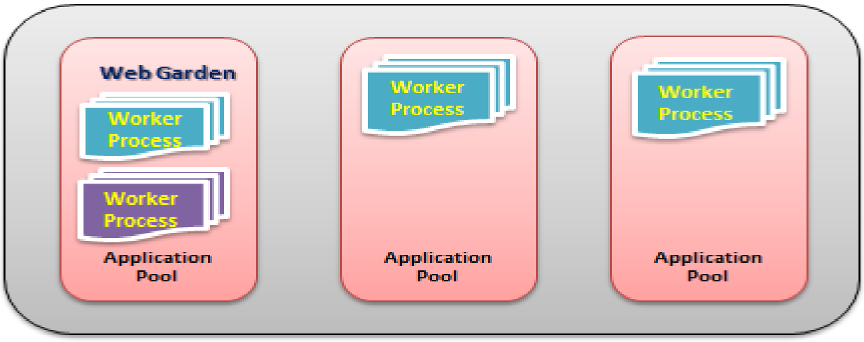


Fig – Multiple worker process in an application pool (Web garden)

Why Web garden?

- Offering applications that are not CPU-bound but execute long running requests the ability to scale and not use up all threads available in the worker process

- Applications that make long running database requests (e.g. high computational database transaction)
- Applications that have threads occupied by long-running synchronous and network intensive transactions
- You can support a greater number of concurrent synchronization operations per CPU by increasing the Maximum Worker Process property value
- Provides better application availability by sharing request between multiple worker processes.
- Web garden use processor affinity where application can swapped out based on preference and tag setting.
- Less consumption of physical space for web garden configuration.

Configure Web garden in IIS 7

Step 1: Deploy and configure your web application as usual
Step 2: Modify web.config file

Find out the following line in web.config file.

```
<sessionState mode="InProc" timeout="60"></sessionState>
```

If the above line exists, remove the above line and replace following line.

```
<sessionState mode="StateServer" cookieless="false" timeout="30"
stateConnectionString="tcpip=127.0.0.1:42424" />
```

Here tcpip = “127.0.0.1” is local state server IP and “42424” is the port number for state server. You can use other IP as state server.

Step 3: Modify registry by regedit

Run →regedt→ HKEY_LOCAL_MACHINE→SYSTEM→CurrentControlSet→Services→aspnet_state→Parameters

Step a: AllowRemoteConnection (Right Click)→ Modify -> Set Value Data to 1 (Base Decimal) →Ok

Step b: Port (Right Click)→ Modify → Base Decimal →Ok

Step 4: Run asp.net state service

Check the following step:

Run→ control→ Administrative Tools→ Services→ ASP.NET State Service (Right Click)→ Properties

Step a: Startup Type→ Automatic

Step b: Service Status→ start

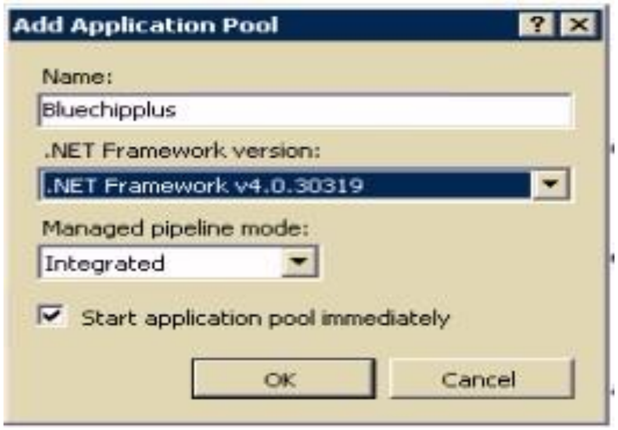
Finally Click Ok.

Step 5: Create an application pool

Create a different application pool for safety. If you use a separate application pool, it will be easy to manage. You can use default application pool also.

To create an application pool that runs using a custom service account

1. Start Internet Information Services (IIS) Manager. (Run->inetmgr)
2. In the left pane, expand the local computer and then select **Application Pools**.
3. Right-click the **Application Pools** node, clicks **Add Application Pool**, and then clicks **Application Pool**.
4. In the **Add Application Pool** dialog box, type **"Pool Name"** in the **name** text box. In figure Pool name is **Bluechipplus**. Select **.NET Framework version.** In figure, I select 4.0.30319. Select **Managed pipeline mode**. I select **integrated mode**. Selection of framework version and managed pipeline mode depends on your application which you are going to hosted. Click OK to complete application pool creation.



Name: Bluechipplus

.NET Framework version: .NET Framework v4.0.30319

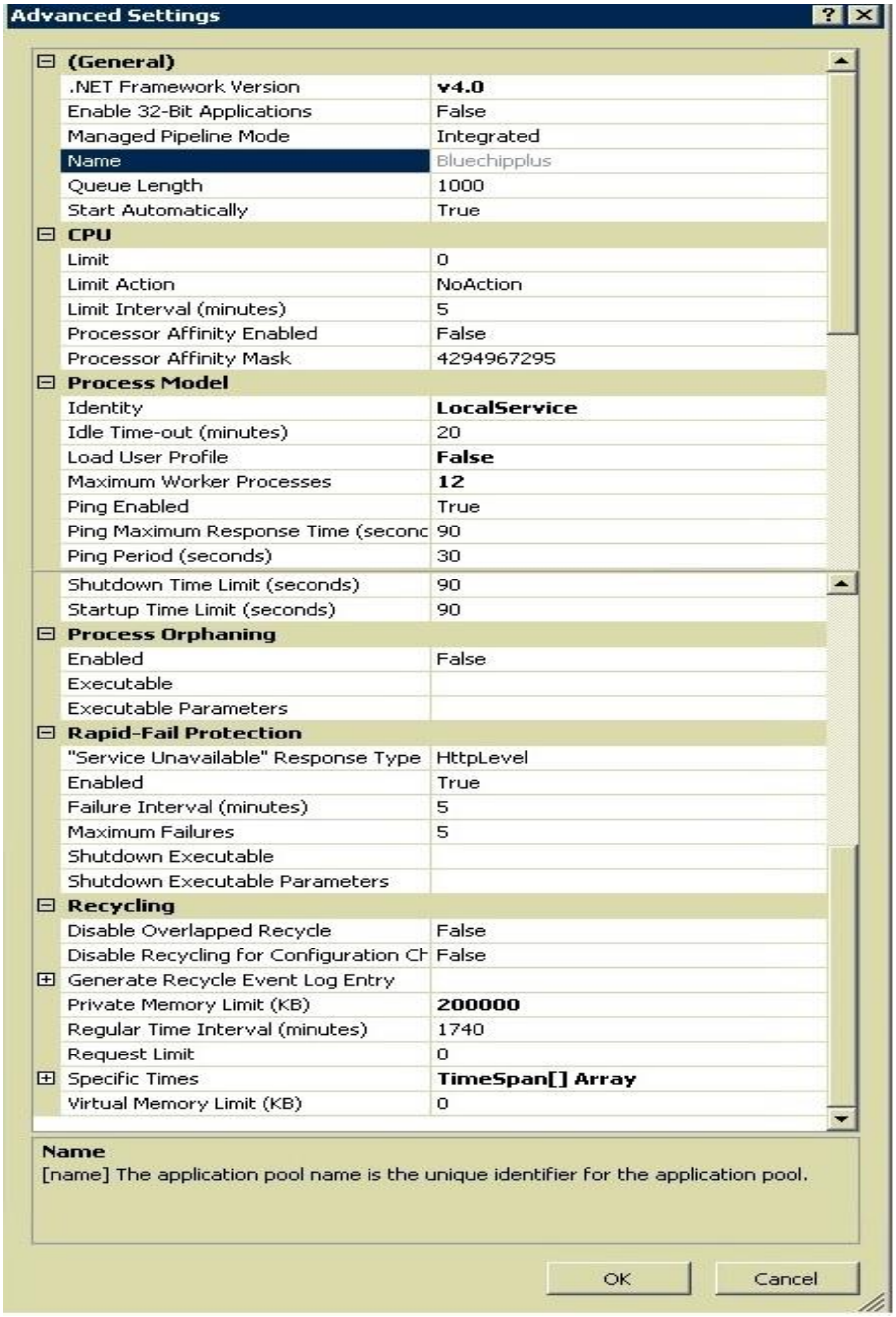
Managed pipeline mode: Integrated

☑ Start application pool immediately

5. Now select application pools. You will see newly created **Bluechipplus** application pool in the middle panel.

Step 6: Configure application pool

1. Select newly created "Bluechipplus" application pool in the middle panel.
2. Click Advanced Settings in "Edit Application Pool" section in the right panel. A dialogue box will appear named "Advanced Settings"
3. In Process Model section, select identity. Network service is default and recommended. If you find problem, you should change. In figure **Local service** is selected.
4. In Process Model section, type maximum worker process. In figure 12 maximum worker processes selected. Selecting worker process is actually depends on your server capability and memory.
5. In Recycling section, select private memory limit in KB. In figure 200000 is selected. Consider your server memory before considering memory limit.
6. Finally click OK to complete application pool settings.

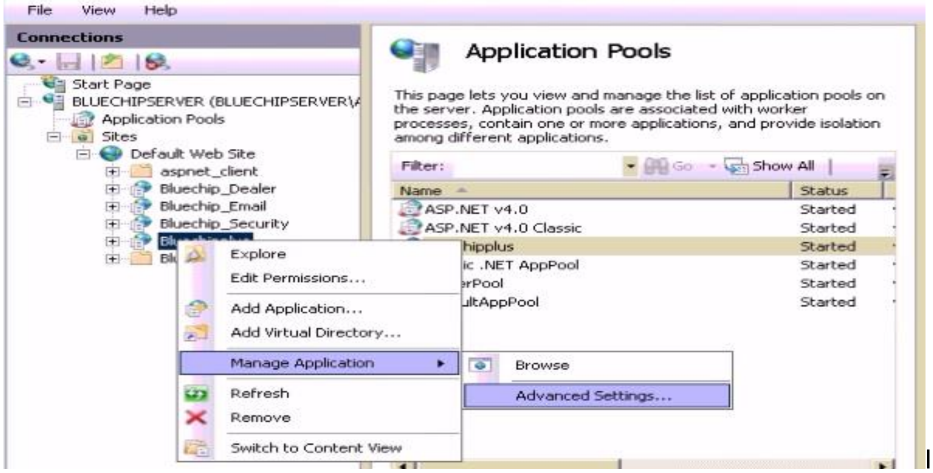


⊟ **(General)**	
.NET Framework Version	**v4.0**
Enable 32-Bit Applications	False
Managed Pipeline Mode	Integrated
Name	Bluechipplus
Queue Length	1000
Start Automatically	True
⊟ **CPU**	
Limit	0
Limit Action	NoAction
Limit Interval (minutes)	5
Processor Affinity Enabled	False
Processor Affinity Mask	4294967295
⊟ **Process Model**	
Identity	**LocalService**
Idle Time-out (minutes)	20
Load User Profile	**False**
Maximum Worker Processes	**12**
Ping Enabled	True
Ping Maximum Response Time (seconc	90
Ping Period (seconds)	30
Shutdown Time Limit (seconds)	90
Startup Time Limit (seconds)	90
⊟ **Process Orphaning**	
Enabled	False
Executable	
Executable Parameters	
⊟ **Rapid-Fail Protection**	
"Service Unavailable" Response Type	HttpLevel
Enabled	True
Failure Interval (minutes)	5
Maximum Failures	5
Shutdown Executable	
Shutdown Executable Parameters	
⊟ **Recycling**	
Disable Overlapped Recycle	False
Disable Recycling for Configuration Ch	False
⊞ Generate Recycle Event Log Entry	
Private Memory Limit (KB)	**200000**
Regular Time Interval (minutes)	1740
Request Limit	0
⊞ Specific Times	**TimeSpan[] Array**
Virtual Memory Limit (KB)	0

Name
[name] The application pool name is the unique identifier for the application pool.

Step 7: Select application pool for the application

1. Right click on application in IIS->Manage Application -> Advanced Settings

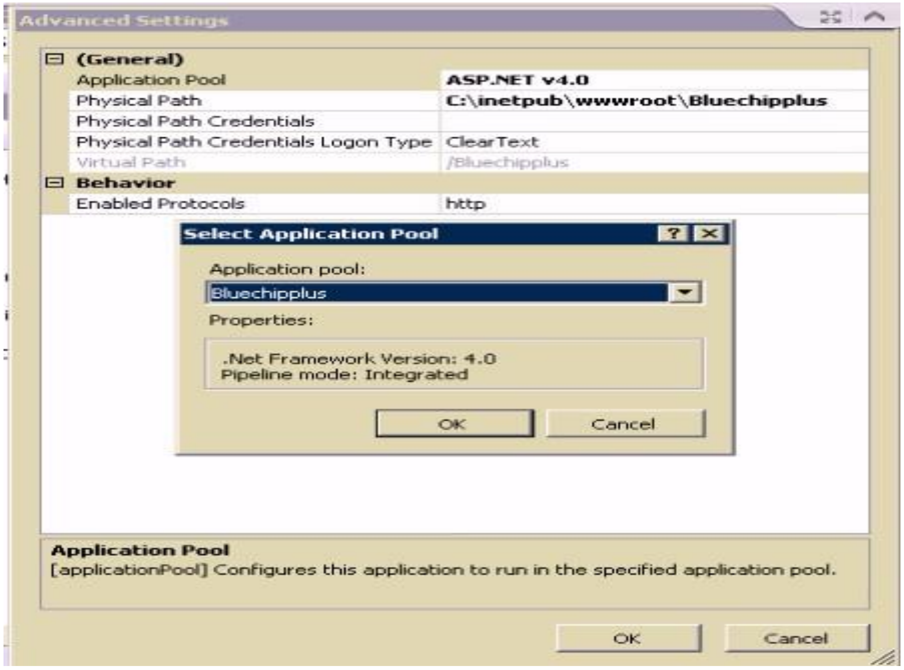


Application Pools

This page lets you view and manage the list of application pools on the server. Application pools are associated with worker processes, contain one or more applications, and provide isolation among different applications.

Name	Status
ASP.NET v4.0	Started
ASP.NET v4.0 Classic	Started
hipplus	Started
ic .NET AppPool	Started
rPool	Started
ultAppPool	Started

2. From advanced settings dialogue box -> Browse application pool -> Select application pool



(General)	
Application Pool	ASP.NET v4.0
Physical Path	C:\inetpub\wwwroot\Bluechipplus
Physical Path Credentials	
Physical Path Credentials Logon Type	ClearText
Virtual Path	/Bluechipplus
Behavior	
Enabled Protocols	http

Select Application Pool

Application pool: Bluechipplus

Properties:

.Net Framework Version: 4.0
Pipeline mode: Integrated

Application Pool
[applicationPool] Configures this application to run in the specified application pool.

3. Finally click OK after assigning application pool

Now you have completed web garden for your application. Browse your application and cheers!!!

Source: http://mahedee.blogspot.com/2013/10/configure-web-gardens-in-iis-7-step-by.html