WCF Deployment

This document records information about the deployment steps I go through at Vincent.

Videos

<http://www.codeproject.com/Articles/550796/A-Beginners-Tutorial-on-How-to-Host-a-WCF-Service>

<http://debugmode.net/2010/09/07/walkthrough-on-creating-wcf-4-0-service-and-hosting-in-iis-7-5/>

<https://msdn.microsoft.com/en-us/library/bb924407.aspx>

<http://gasparnagy.com/2014/01/enable-net-4-0-wcf-hosing-in-iis-8/>

<http://www.topwcftutorials.net/2013/09/simple-steps-for-restful-service.html>

<http://stackoverflow.com/questions/24530411/how-can-i-host-a-wcf-service-without-an-svc-file-in-iis>

<http://forums.iis.net/t/1214419.aspx?How+to+remove+svc+from+wcf+rest+service>

<http://www.codemag.com/Article/0701041>

I have been able to build the rest service and deploy it to my local iis server. For the service with the IBM libraries I had to adjust the application pool settings.

<https://d4dilip.wordpress.com/2013/02/14/could-not-load-file-or-assembly-xxxxx-or-one-of-its-dependencies-an-attempt-was-made-to-load-a-program-with-an-incorrect-format/>

The following links I found and thought it a good idea to keep them around.

Four new features that can be helpful, the logging and trace features are great.

<http://www.dotnetcurry.com/showarticle.aspx?ID=990>

## Deploy the Service Implementation to the IIS Application

WCF services hosted in IIS use the same dynamic compilation model as [ASP.NET](http://asp.net/) 2.0. Just as with[ASP.NET](http://asp.net/), you can deploy the implementation code for IIS-hosted WCF services in several ways at various locations, as follows:

* As a precompiled .dll file located in the global assembly cache (GAC) or in the application’s \bin directory. Precompiled binaries are not updated until a new version of the class library is deployed.
* As un-compiled source files located in the application’s \App\_Code directory. Source files located in this directory are dynamically required when processing the application’s first request. Any changes to files in the \App\_Code directory cause the entire application to be recycled and recompiled when the next request is received.
* As un-compiled code placed directly in the .svc file. Implementation code can also be located inline in the service’s .svc file, after the @ServiceHost directive. Any changes to inline code cause the application to be recycled and recompiled when the next request is received.

For more information about the [ASP.NET](http://asp.net/) 2.0 compilation model, see [ASP.NET Compilation Overview](http://go.microsoft.com/fwlink/?LinkId=94773).

## Configure the WCF Service

IIS-hosted WCF services store their configuration in the applications Web.config file. IIS-hosted services use the same configuration elements and syntax as WCF services hosted outside of IIS. However, the following constraints are unique to the IIS hosting environment:

* Base addresses for IIS-hosted services.
* Applications hosting WCF services outside of IIS can control the base address of the services they host by passing a set of base address URIs to the [ServiceHost](https://msdn.microsoft.com/en-us/library/system.servicemodel.servicehost.aspx) constructor or by providing a [<host>](https://msdn.microsoft.com/en-us/library/ms789018.aspx) element in the service’s configuration. Services hosted in IIS do not have the ability to control their base address; the base address of an IIS-hosted service is the address of its .svc file.

### Aa751792.collapse_all(en-us,VS.110).gifEndpoint Addresses for IIS-Hosted Services

When hosted in IIS, endpoint addresses are always considered to be relative to the address of the .svc file that represents the service. For example, if the base address of a WCF service is <http://localhost/Application1/MyService.svc> with the following endpoint configuration.

<endpoint address="anotherEndpoint" .../>

This provides an endpoint that can be reached at "<http://localhost/Application1/MyService.svc/anotherEndpoint>".

Similarly, the endpoint configuration element that uses an empty string as the relative address provides an endpoint reachable at <http://localhost/Application1/MyService.svc>, which is the base address.

<endpoint address="" ... />

You must always use relative endpoint addresses for IIS-hosted service endpoints. Supplying a fully-qualified endpoint address (for example, <http://localhost/MyService.svc>) can lead to errors in the deployment of the service if the endpoint address does not point to the IIS-application that hosts the service exposing the endpoint. Using relative endpoint addresses for hosted services avoids these potential conflicts.

### Aa751792.collapse_all(en-us,VS.110).gifAvailable Transports

WCF services hosted in IIS 5.1 and IIS 6.0 are restricted to using HTTP-based communication. On these IIS platforms, configuring a hosted service to use a non-HTTP binding results in an error during service activation. For IIS 7.0, the supported transports include HTTP, Net.TCP, Net.Pipe, Net.MSMQ, and msmq.formatname for backwards compatibility with existing MSMQ applications.

### Aa751792.collapse_all(en-us,VS.110).gifHTTP Transport Security

IIS-hosted WCF services can make use of HTTP transport security (for example, HTTPS and HTTP authentication schemes such as Basic, Digest, and Windows Integrated Authentication) as long as the IIS virtual directory that contains the service supports those settings. The HTTP Transport Security settings on a hosted endpoint’s binding must match the transport security settings on the IIS virtual directory that contains it.

For example, a WCF endpoint configured to use HTTP digest authentication must reside in an IIS virtual directory that is also configured to allow HTTP digest authentication. Unmatched combinations of IIS settings and WCF endpoint settings result in an error during service activation.