**Hosting in WCF**

**Table of Contents**

[Hosting 2](#_Toc451603321)

[Needs of Hosting 3](#_Toc451603322)

[Ways to Host 3](#_Toc451603323)

[Service Host 4](#_Toc451603324)

[Self Hosting 6](#_Toc451603325)

[Step 0: 6](#_Toc451603326)

[Step 1: 7](#_Toc451603327)

[Summary 9](#_Toc451603328)

# 

# WAS Hosting

WAS => Windows Activation Service

WAS is a Process Activation Service comes with Windows vista and IIS 7. WAS was designed to support non-HTTP protocols with the help of IIS 7 where as IIS 6 supports only HTTP protocols. If you want to use NetTcpBinding over IIS, WAS hosting is the only option. Of course, WAS also supports HTTP protocols

# Needs of Hosting

Once you developed your service you need make it available to the client, then only the client can

access it and get the benefits, In our case we can call it as "Online Shopping Service", Benefit what client

gets: Purchasing an item from an online store using the  mobile app. The service is now ready with a full

set of online shopping service features like "Purchase Item", "Sell Item". Then we need to host it or place

it in a position. This process is called hosting, it creates various configuration settings that enable the

service execution, prepares the environment, creates the endpoint, starts the listening process and

manages the service lifecycle

# Ways to Host

There are few ways to hosting your service to be accessible across the world.

1. Self Hosting => Using Windows Console and Form Application
2. IIS Hosting => Using IIS6
3. WAS Hosting => Using IIS7
4. Windows Service Application

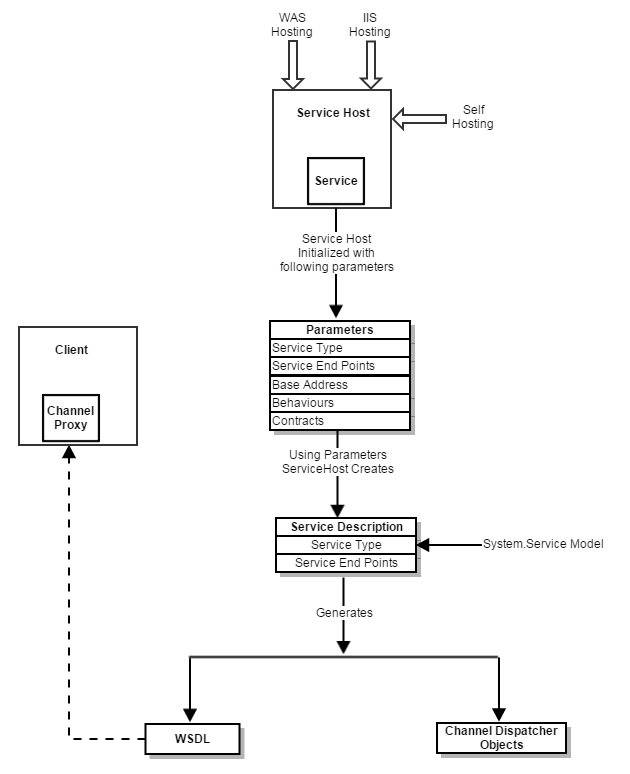
Selecting the ways of hosting is depending upon the transport protocol and the feature you expect in the hosting environment, There are many more features will be needed to make the service a robust host. Few important features like Health Monitoring, Process Recycling,Process Activation and Idle Time Management are highly recommended to host the service.I will not detail out the feature here but i will brief out.

 Process Recycling - If the IIS finds the service is not healthy enough due to the memory leak or other factors ,it will restart the worker process and avoiding the entire system to get into the problem

 Process Activation - Worker process or other processes will be triggered only the request arrives. Unlike SelfHosting ,the host process should always be started up and listening for incoming requests. There is always a big debate on Which hosting will be better for your service? of -course, it's an interesting to discuss.But i don't want to hijack the topic, we will discuss this in the series of my articles in hosting.

# Service Host

Service Host is a part of System.ServiceModel Namespace. ServiceHost instance associates one or more services to be accessible at run- time. when you are trying to do hosting in a console application(self-hosting), then you need to use the class Service Host. If you are trying to host your service in WAS or IIS,  WAS or IIS will use the instance ServiceHost on behalf of you. ServiceHost will be initialized with the parameters ServiceType,Service EndPoints,Base Address and Behaviour and creates a service description that is a part of System.ServiceModel. The ServiceDescription is responsible for creating the description of the service that includes the types of service,endpoint and behaviours, and uses that description to create the channel dispatcher objects and WSDL document so that the client can access the exposed services. The following diagram shows the process of service host

[](https://techcarvings.files.wordpress.com/2015/06/service-host.png)

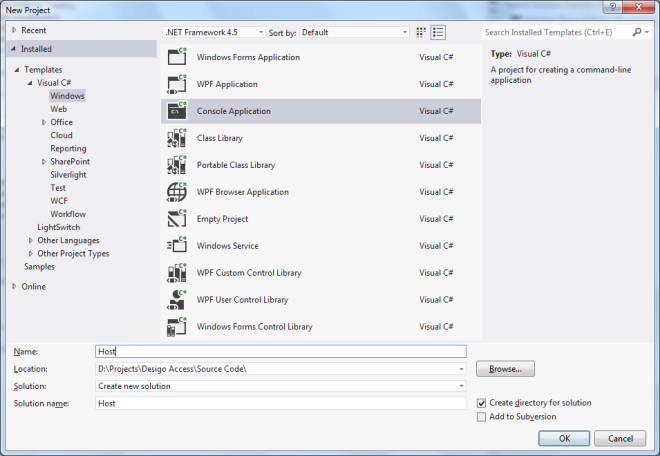
The  diagram above tells exactly what i have explained you in the sub-topic Service Host.

# Self Hosting

The simplest way to host your services is self-hosting that can be suffice using the Winforms,Console applications and windows services. The Developer must simply instantiate the Service Host class that is derived from system.servicemodel. Service Host will be opened when it will be instantiated with the parameters service type,contract type and endpoints.Once it is opened it will allocate the work process to receive the incoming messages from the client and served it. Ok ,Let's move on to discuss the steps required for the Self Hosting.

# Step 0:

In the First step, you can find out that New Host Project will be created. For the Demonstration purpose ,i have selected the console application in windows templates. you can also select the Host project as Windows Form application or WPF application. The following snapshot provides a clear picture on first step.

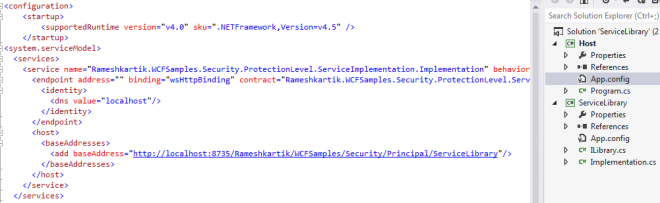


# Step 1:

In the second step, you can find out that ServiceHost is instantiated with the service type. In our Demonstration ,you can find that the ServiceType parameter(Implementation) is passed to the ServiceHost class. This call will create the service to communicate in the communication protocol mentioned(wsHttpBinding) with the DataContract ILibrary to host in the address mentioned in the BaseAddress



You can also find out that Event handlers have been assigned to the service host class and it will be fired if the ServiceHost is opened or closed appropriately. Please ensure that servicetype(implementation) declaration is happened since it mentioned in the Host App.config file, You can also do it programmatically



Here with I have attached the SourceCode as well,you can have a look at that. Self Hosting is very simple and you don't require too many steps to host it. Yes, That's it, am going to end this article, thought of giving a detailed explanation on WAS Hosting and IIS Hosting in my next article.

# Summary

Service Host is a part of System.ServiceModel .This class will be instantiated by the developer if he needs to do SelfHosting. This class will be instantiated by IIS if the service is hosted in the way of IIS or WAS.