

Does WCF raise the bar or just the complexity level? [closed]

Asked 16 years, 3 months ago Modified 5 years, 7 months ago

Viewed 13k times



84



Closed. This question is [opinion-based](#). It is not currently accepting answers.



Want to improve this question? Update the question so it can be answered with facts and citations by [editing this post](#).

Closed 4 years ago.

[Improve this question](#)

I understand the value of the three-part service/host/client model offered by WCF. But is it just me or does it seem like WCF took something pretty direct and straightforward (the ASMX model) and made a mess out of it?

Is there an alternative to using SvcUtil's command line step back in time to generate the proxy? With ASMX services a test harness was automatically provided; is there a good alternative today with WCF?

I appreciate that the WS* stuff is more tightly integrated with WCF and hope to find some payoff for WCF there,

but geeze, otherwise I'm perplexed.

Also, the state of books available for WCF is abysmal at best. Juval Lowy, a superb author, has written a good O'Reilly reference book "Programming WCF Services" but it doesn't do that much (for me anyway) for learning now to use WCF. That book's precursor (and a little better organized, but not much, as a tutorial) is Michele Leroux Bustamante's Learning WCF. It has good spots but is outdated in place and its corresponding Web site is gone.

Do you have good WCF learning references besides just continuing to Google the bejebus out of things?

wcf

web-services

Share

Improve this question

Follow

edited Apr 27, 2019 at 7:13



grooveplex

2,529 ● 4 ● 29 ● 30

asked Sep 8, 2008 at 16:42



rp.

17.6k ● 14 ● 65 ● 79

-
- 3 If you're looking for good test client for WCF services. Look no further than msdn.microsoft.com/en-us/library/bb552364.aspx – Strelak Dec 10, 2008 at 22:35
-

16 Answers

Sorted by:

Highest score (default)





61



Okay, here we go. First, Michele Leroux Bustamante's book has been updated for VS2008. The website for the book is not gone. It's up right now, and it has tons of great WCF info. On that website she provides updated code compatible with VS2008 for all the examples in her book. If you order from Amazon, you will get the reprint which is updated.

WCF is not *only* a replacement for ASMX. Sure it can (and does quite well) replace ASMX, but the real benefit is that it allows your services to be self-hosted. Most of the functionality from WSE has been baked in from the start. The framework is *highly* configurable, and the ability to serve multiple endpoints over multiple protocols is amazing, IMO.

While you can still generate proxy classes from the "Add Service Reference" option, it's not necessary. All you really have to do is copy your ServiceContract interface and tell your code where to find the endpoint for the service, and that's it. You can call methods from the service with very little code. Using this method, you have complete control over the implementation. Regardless of the method you choose to generate a proxy class, Michele shows both and uses both in her **excellent** series of webcasts on the subject.

Michele has tons of great material out there, and I recommend you check out her website(s). Here's some links that were incredibly helpful for me as I was learning WCF. I hope that you'll come to realize how strong WCF

really is, and how easy it is to implement. The learning curve is a little bit steep, but the rewards for your time investment are well worth it:

- Michele's webcasts:
<http://www.dasblonde.net/2007/06/24/WCFWebcastSeries.aspx>
- Michele's book website (alive and updated for VS2008): <http://www.thatindigogirl.com/>

I recommend you watch at least 1 of Michele's webcasts. She is a very effective presenter, and she's obviously incredibly knowledgeable when it comes to WCF. She does a great job of demystifying the inner workings of WCF from the ground up.

Share Improve this answer

edited Jan 15, 2010 at 14:48

Follow

answered May 18, 2009 at 13:04



Scott Arrington

12.5k ● 3 ● 44 ● 55

-
- 3 Great resources. I wouldn't go as far as to say rp was bashing WCF or the authors. I too have felt a little overwhelmed while attempting to look at WCF after having spent so much time in ASMX. I know it's better and I want to experience it, but it's hard to find an entry point quite like you could with ASMX. – [Chris Stewart](#) Aug 11, 2009 at 0:59
-



I typically use Google to find my WCF answers and commonly find myself on the following blogs:

15



Blogs with valuable WCF articles



- <http://blogs.msdn.com/drnick/default.aspx>
- <http://blogs.msdn.com/wenlong/default.aspx>
- <http://blogs.thinktecture.com/buddhike/>
- <http://www.dasblonde.net/default.aspx>

Other valuable articles I've found

- <http://blogs.conchango.com/pauloreichert/archive/2007/02/22/WCF-Reliable-Sessions-Puzzle.aspx>
- <http://blogs.msdn.com/salvapatuel/archive/2007/04/25/why-using-is-bad-for-your-wcf-service-host.aspx>

Share Improve this answer

answered Dec 10, 2008 at 22:53

Follow



Chris Porter

3,687 ● 27 ● 29

Thanks for the links. Just curious, do you know what happened to Buddhike? He seems to have disappeared off the face of the earth. The new blog he mentions at blogs.thinktecture.com/buddhike doesn't exist (anymore). It's like he's shuffled off this mortal coil. (What dreams may come?) Thanks. – Jim Raden Sep 2, 2009 at 22:08

blogs.conchango.com/pauloreichert/archive/2007/02/22/... not found – PreguntonCojoneroCabrón Dec 23, 2017 at 23:17



14



I'm having a hardtime to see when I should or would use WCF. Why? Because I put productivity and simplicity on top of my list. Why was the ASMX model so succesful, because it worked, and you get it to work fast. And with VS 2005 and .NET 2.0 wsdl.exe was spitting out pretty nice and compliant services.

In real life you should have very few communication protocols in your architecture. This keeps it simple an maintainable. If you need to acces to legacy systems, write specific adapters for them so they can play along in the nice shiny and beautiful SOA world.

Share Improve this answer

answered Dec 8, 2008 at 16:14

Follow



Saab

1,001 ● 3 ● 13 ● 37

WCF aims to be a single framework for all those protocols. It does SOAP, REST, and with the WSCF Adapter SDK, it does those "legacy system" connections too, all within the WCF model. – [Cheeso](#) May 18, 2009 at 12:34

3 It depends on how you measure "productivity". I would rather a developer take 2 - 3 days to get their head around this now, for the benefits say in 6 months time. WCF replaces more than just webservices. – [Christian Payne](#) Jun 21, 2009 at 0:07

1 I agree 100%. If you just want Web Services and you need to ship quickly (who doesn't?), its ASMX all the way baby! WCF is a giant sledgehammer - total overkill if all you want is basic Web Services. Just because its the shiny new thing does not mean you should assume its all round better, for all

scenarios, all the time. And just because WCF is hard to understand, doesn't make you a smart person for choosing it. More power to those with the guts to decline to jump onto the WCF bandwagon! – [saille](#) Jul 14, 2010 at 0:34



13



WCF is much more powerful than ASMX and it extends it in several ways. ASMX is limited to only HTTP, whereas WCF can use several protocols for its communication (granted, HTTP is still the way most people will use it, at least for services that need to be interoperable). WCF is also easier to extend. At least, it is possible to extend it in ways that ASMX cannot be extended. "Easy" may be stretching it. =)

The added functionality offered by WCF far outweighs the complexity it adds, in my opinion. I also feel that the programming model is easier. DataContracts are much nicer than having to serialize using XML serialization with public properties for everything, for example. It's also much more declarative in nature, which is also nice.

Share Improve this answer

answered Sep 26, 2008 at 10:29

Follow



[Karl](#)

467 ● 2 ● 11



6

Wait.... did you ever use .NET Remoting, cause thats the real thing its replacing. .NET Remoting is pretty complicated itself. I find WCF easier and better laid out.

Share Improve this answer

answered Sep 8, 2008 at 17:13



Follow



Quibblesome

25.4k ● 10 ● 62 ● 104



4

I don't see it mentioned often enough, but you **can** still implement fairly simple services with WCF, very similar to ASMX services. For example:



```
[ServiceContract]
[AspNetCompatibilityRequirements(RequirementsMode =
AspNetCompatibilityRequirementsMode.Allowed)]
public class SimpleService
{
    [OperationContract]
    public string HelloWorld()
    {
        return "Hello World";
    }
}
```

You still have to register the end point in your web.config, but that's not so bad.

Eliminating the verbosity of the separated data, service, and operation contracts goes a long way toward making WCF more manageable for me.

Share Improve this answer

answered Sep 8, 2008 at 16:53

Follow



Dave Ward

60.5k ● 14 ● 118 ● 134

AspNetCompatibilityRequirements is a necessary evil if many developers are going to make the switch from their ASMX services which are currently working just fine. – [Dave Ward](#)
Mar 23, 2009 at 15:17

@dan why not **AspNetCompatibilityRequirements** ?
– [PreguntonCojoneiroCabrón](#) Dec 23, 2017 at 23:19

@DaveWard **AspNetCompatibilityRequirements is a necessary evil** ? – [PreguntonCojoneiroCabrón](#) Dec 23, 2017 at 23:20



4

VS2008 includes the "Add Service Reference" context menu item which will create the proxy for you behind the scenes.



As was mentioned previously, WCF is not intended solely as a replacement for the ASMX web service types, but to provide a consistent, secure and scalable methodology for all interoperable services, whether it is over HTTP, tcp, named pipes or MSMQ transports.



I will confess that I do have other issues with WCF (e.g. re-writing method signatures when exposing a service over basicHTTP - see [here](#), but overall I think it is a definite improvement

Share Improve this answer

Follow

edited May 23, 2017 at 10:29



Community Bot

1 • 1

answered Sep 26, 2008 at 10:45



ZombieSheep

29.9k ● 12 ● 70 ● 114



3

If you're using VS2008 and create a WCF project then you automatically get a test harness when you hit run/debug and you can add a reference without having to use svcutil.



Share Improve this answer

answered Sep 26, 2008 at 10:33



Follow



blowdart

56.5k ● 12 ● 118 ● 151



2

My initial thoughts of WCF were exactly the same! Here are some solutions:



1. Program your own proxy/client layer utilising generics (see classes [ClientBase](#), Binding). I've found this easy to get working, but hard to perfect.
2. Use a third party implementation of 1 ([SoftwareIsHardwork](#) is my current favourite)



Share Improve this answer

answered Dec 10, 2008 at 23:10

Follow



NoizWaves

2,680 ● 6 ● 28 ● 32



2

WCF is a replacement for all earlier **web service** technologies from Microsoft. It also does a lot more than what is traditionally considered as "web services".



WCF "web services" are part of a much broader spectrum of remote communication enabled through WCF. You will get a much higher degree of flexibility and portability doing things in WCF than through traditional ASMX because WCF is designed, from the ground up, to summarize all of the different distributed programming infrastructures offered by Microsoft. An endpoint in WCF can be communicated with just as easily over SOAP/XML as it can over TCP/binary and to change this medium is simply a configuration file mod. In theory, this reduces the amount of new code needed when porting or changing business needs, targets, etc.

ASMX is older than WCF, and anything ASMX can do so can WCF (and more). Basically you can see WCF as trying to logically group together all the different ways of getting two apps to communicate in the world of Microsoft; ASMX was just one of these many ways and so is now grouped under the WCF umbrella of capabilities.

Web Services can be accessed only over HTTP & it works in stateless environment, where WCF is flexible because its services can be hosted in different types of applications. Common scenarios for hosting WCF services are IIS, WAS, Self-hosting, Managed Windows Service.

The major difference is that Web Services Use XmlSerializer. But WCF UsesDataContractSerializer

which is better in Performance as compared to XmlSerializer.

In what scenarios must WCF be used

- A secure service to process business transactions. A service that
- supplies current data to others, such as a traffic report or other
- monitoring service. A chat service that allows two people to
- communicate or exchange data in real time. A dashboard application
- that polls one or more services for data and presents it in a logical
- presentation. Exposing a workflow implemented using Windows Workflow
- Foundation as a WCF service. A Silverlight application to poll a
- service for the latest data feeds.

Features of WCF

- Service Orientation
- Interoperability
- Multiple Message Patterns
- Service Metadata
- Data Contracts

- Security
- Multiple Transports and Encodings
- Reliable and Queued Messages
- Durable Messages
- Transactions
- AJAX and REST Support
- Extensibility

source: [main source of text](#)

Share Improve this answer

answered May 5, 2014 at 21:25

Follow



[Babak.Abad](#)

2,936 ● 11 ● 41 ● 76

Now, use WCF or REST? or WebAPI ? or MicroService ?

– [PreguntonCojonoCabrón](#) Dec 23, 2017 at 23:18



MSDN? I usually do pretty well with the Library reference itself, and I usually expect to find valuable articles there.

1



Share Improve this answer

answered Sep 8, 2008 at 16:45

Follow



[Ben Collins](#)

20.7k ● 18 ● 128 ● 190





1



In terms of what it offers, I think the answer is compatibility. The ASMX services were pretty Microsofty. Not to say that they didn't try to be compatible with other consumers; but the model wasn't made to fit much besides ASP.NET web pages and some other custom Microsoft consumers. Whereas WCF, because of its architecture, allows your service to have very open-standard--based endpoints, e.g. REST, JSON, etc. in addition to the usual SOAP. Other people will probably have a much easier time consuming your WCF service than your ASMX one.

(This is all basically inferred from comparative MSDN reading, so someone who knows more should feel free to correct me.)

Share Improve this answer

answered Sep 8, 2008 at 17:11

Follow



Domenic

113k ● 42 ● 226 ● 273

I had no trouble with cross-platform ASMX services--XML does quite nicely there. I understand that WCF = remoting + ASMX + MSMQ + WSE. My beef is that MS should use a little "progressive enhancement" to make WCF more easily approachable and my question is how! Thanks, rp – [rp.](#) Sep 8, 2008 at 17:17

ASMX interoperability was very good, though the effort got higher when WS-Security was involved. – [Cheeso](#) May 18, 2009 at 12:36



1



WCF should not be thought of as a replacement for ASMX. Judging at how it is positioned and how it is being used internally by Microsoft, it is really a fundamental architecture piece that is used for any type of cross-boundary communication.



Share Improve this answer

answered Dec 8, 2008 at 19:52



Follow

 Kwal

 1,531 ● 7 ● 11

yes...and as a fundamental communications framework, it should replace ASMX and ASP.NET-oriented web services.

No? – [Cheeso](#) May 18, 2009 at 12:37



1



I believe that WCF really advances ASMX web services implementation in many ways. First of all it provides a very nice layered object model that helps hide the intrinsic complexity of distributed applications. Secondly you can have more than request-replay messaging patterns, including asynchronous notifications from server to client (impossible with pure HTTP), and thirdly abstracting away the underlying transport protocol from XML messaging and thus elegantly supporting HTTP, HTTPS, TCP and other. Backward compatibility with "1-st generation" web services is also a plus. WCF uses XML standard as the internal representation format. This could be perceived as advantage or disadvantage, especially with the growing popularity "fat-free alternatives to XML" like JSON.



Share Improve this answer

answered Dec 8, 2008 at 20:13

Follow



[pglowack](#)

291 ● 4 ● 9



1

The difficult things I find with WCF is managing the configurations for clients and servers, and troubleshooting the not so nice faulted state exceptions.



It would be great if anyone had any shortcuts or tips for those.



Share Improve this answer

answered [Dec 31, 2008 at 20:16](#)

Follow

community wiki

[StingyJack](#)



1

I find that is a pain; in that I have .NET at both ends, have the same "contract" dlls loaded at both ends etc. But then I have to mess about with a lot of details like "KnownType" attributes.



WCF also defaults to only letting 1 or 2 clients connect to a service until you change lots of configuration. Changing the config from code is not easy, shipping lots of config files is not an option, as it is too hard to merge our changes into any changes a customer may have made at the time of an upgrade (also we don't want customers playing with WCF settings!)



.NET remoting tended to just work most of the time.

I think trying to **pretend** that .NET to .NET object based communications is the same as sending bit so of Text (xml) to an unknown system, was a step too far.

(The few times we have used WCF to talk to a Java system, we found that the XSD that the java system gave out did not match what XML it wanted anyway, so had to hand-code a lot of the XML mappings.)

Share Improve this answer

edited Mar 28, 2010 at 14:27

Follow

answered Mar 11, 2010 at 9:09



[Ian Ringrose](#)

51.9k ● 57 ● 217 ● 321

I think people are a little off the mark when they say that WCF is a replacement for remoting. Remoting is still a fine option for when you want to do real .Net to .Net object-based calls, and have plenty of control over how things are set up on both ends. WCF on the other hand is geared towards publishing Services for *anyone* to consume, any way they want to. – [Tim Lovell-Smith](#) Jun 23, 2010 at 5:20

@Tim, Microsoft now says use WCF for everything and that it is a replacment for Remoting. Remoting seems to be the forgotten child of the .net world. – [Ian Ringrose](#) Jun 24, 2010 at 8:19
