

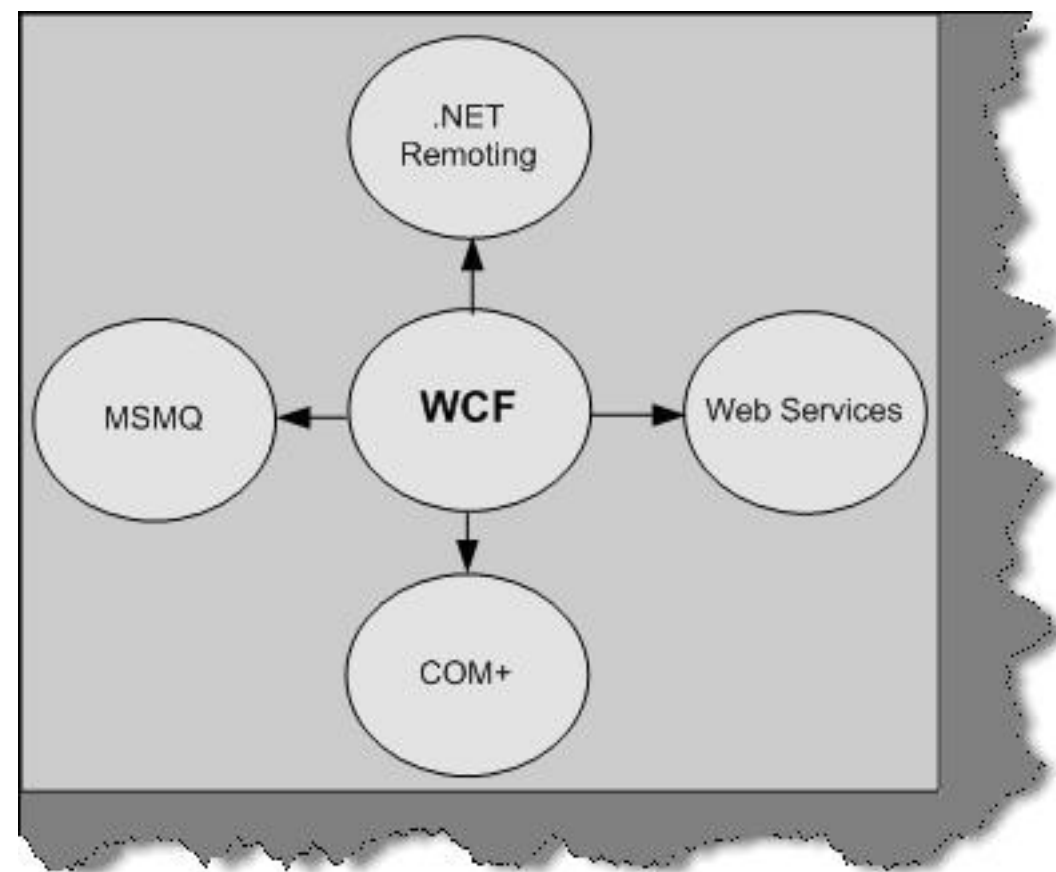
NET: Windows Communication Foundation

[Stampa](#)
(10) » [NET: Upload documenti](#) » [NET: Utilizzo javascript in .NET](#) » **[NET: Windows Communication Foundation](#)**

Indice dei Contenuti [[Nascondi/Mostra](#)]

[Scenario](#)
[Overview, Strumenti](#)
[Specifiche](#)
[Sicurezza](#)
[Esempi](#)
[Note e soluzioni ad alcuni problemi](#)
[WCF Test Client \(WcfTestClient.exe\)](#)
[Configuration Editor Tool \(SvcConfigEditor.exe\)](#)
[Riferimenti](#)

Scenario

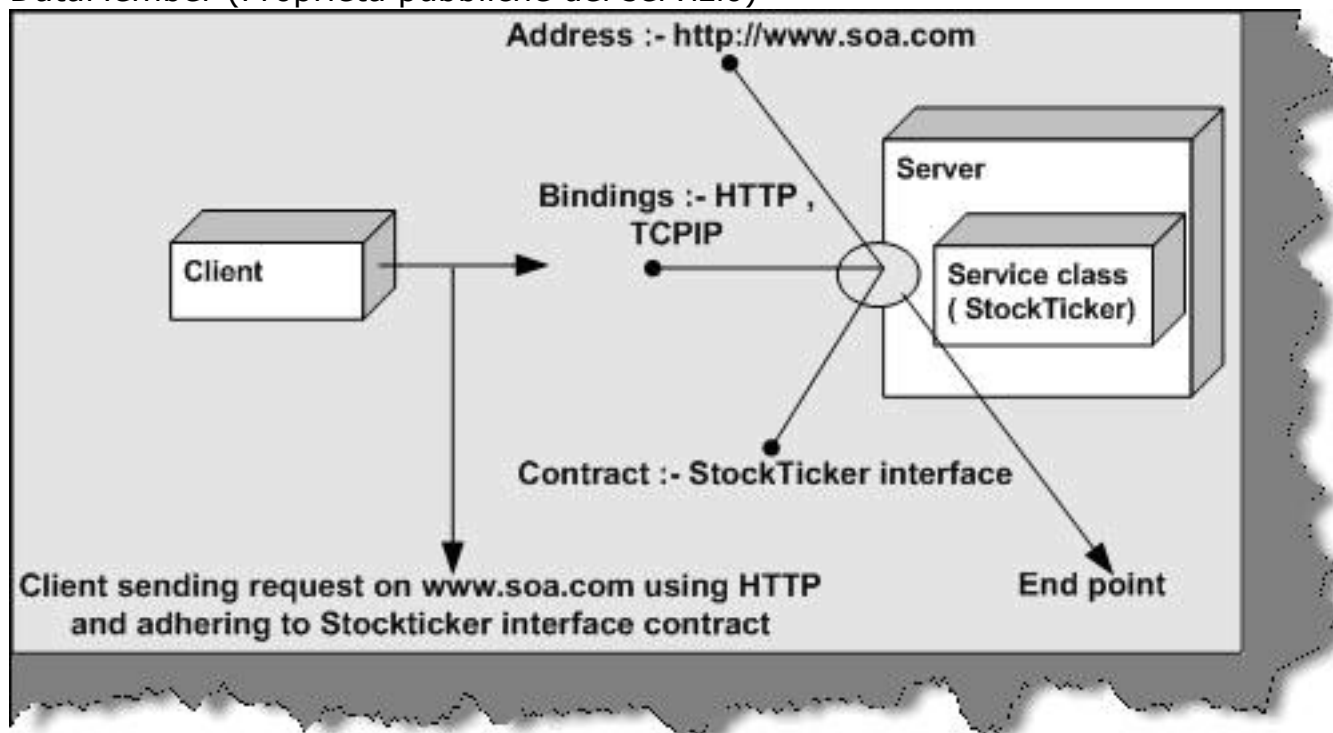


WCF Components

- Nel corso degli anni sono state sviluppate molte tecnologie con lo scopo di realizzare applicazioni distribuite, in grado di comunicare via rete attraverso lo scambio di messaggi. Grazie a queste tecnologie, un'applicazione (definita **Client**) può scambiare messaggi o ricevere dati e risorse da un apposito **servizio**, raggiungibile dalla rete grazie ad un apposito protocollo di comunicazione (tcp, http, ...).
- Nel mondo Microsoft sono state sviluppate diverse tecnologie: Microsoft Messaging Queue, Web Services, .NET Remoting. Queste tecnologie hanno un **limite: un'applicazione può comunicare con i servizi solo se entrambi utilizzano la stessa tecnologia**. Windows Communication Foundation (**WCF**) si propone come un **modello unificato** di programmazione per **tutte le applicazioni distribuite**, che raggruppa tutte le funzionalità di trasmissione dei dati, permettendone la facilità di realizzazione lato client e lato server.

Overview, Strumenti

- In WCF lo scambio di messaggi tra applicazione client e servizio avviene tramite i cosiddetti **Endpoints**.
- L'**ABC** di WCF: ogni Endpoint è suddiviso in tre parti fondamentali.
 - **Address** (Dove): l'indirizzo a cui si trova il servizio, ad esempio un URI
 - **Binding** (Come): è la vera punta di diamante del WCF, infatti si preoccupa di trasformare e trasmettere le nostre informazioni nel protocollo da noi prescelto: http, https, tcp, ecc. In questo modo possiamo sviluppare il nostro servizio senza preoccuparci del modo in cui comunicherà; il Binding inoltre ha il compito di salvaguardare la sicurezza del messaggio, dichiarando dei Behavior per la trasformazione delle nostre informazioni in formato sicuro
 - **Contract** (Cosa): nel Contract vengono definiti i metodi e le proprietà che fanno parte del nostro servizio; per descriverne la struttura è necessario dichiarare una interfaccia che, come un vero contratto, dovrà indicare quali informazioni e quali operazioni saranno rese accessibili ai client che interrogheranno il servizio utilizzando degli attributi, nello specifico:
 - OperationContract (Metodi del servizio utilizzabili dai client)
 - DataContract (Strutture dati come classi, enum e struct)
 - DataMember (Proprietà pubbliche del servizio)



WCF Endpoints

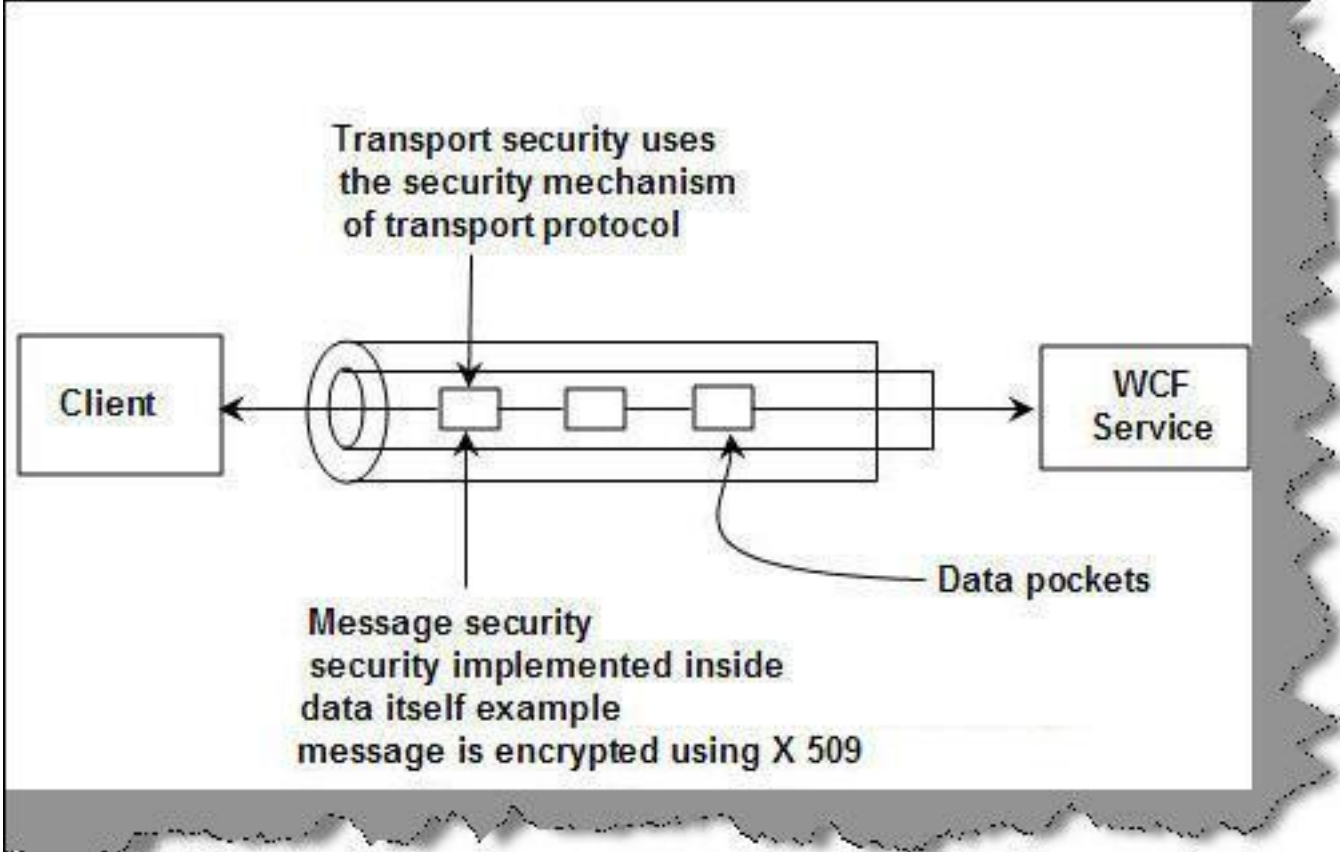
- Si rimandano gli argomenti più articolati come le **Callback** (situazioni in cui è il servizio a dover inviare messaggi al client, come ad esempio in applicazioni di chat o messaggistica istantanea) e la **Sicurezza** ai Riferimenti più sotto.
- Si può utilizzare come IDE sia Visual Studio .NET 2005 sia **Visual Studio .NET 2008**. Per il 2005 è necessario installare sul sistema operativo il Windows SDK, il 2008 invece è già pronto per lo sviluppo con WCF.

Specifiche

- WCF supporta le specifiche WS-* (definite da Microsoft, IBM, SUN e altre grandi compagnie)

Sicurezza

- WCF si occupa delle seguenti 4 "feauteres" riguardanti la sicurezza:
 - **Confidentiality**
 - **Integrity**
 - **Authentication**
 - **Authorization**
- WCF ha la capacità di applicare la sicurezza a livello sia del trasporto (i.e. protocol level) sia del messaggio (i.e. data).



Esempi

- Esempio di porzione di web.config dove è presente la configurazione del servizio WCF ospitato da applicazione WEB ASP.NET

```

1  <system.serviceModel>
2  <behaviors>
3  <serviceBehaviors>
4  <behavior name="myOracleServiceBehavior">
5  <!-- To avoid disclosing metadata information, set the value below to false and remove the
6  <serviceMetadata httpGetEnabled="true" />
7  <!-- To receive exception details in faults for debugging purposes, set the value below to
8  <serviceDebug includeExceptionDetailInFaults="false" />
9  </behavior>
10 </serviceBehaviors>
11 </behaviors>
12 <services>
13 <service behaviorConfiguration="myOracleServiceBehavior" name="myOracleService">
14 <!-- Service Endpoints -->
15 <endpoint address="" binding="wsHttpBinding" contract="ImyOracleService">
16 <!--
17 Upon deployment, the following identity element should be removed or replaced to reflect the
18 identity under which the deployed service runs. If removed, WCF will infer an appropriate
19 identity automatically.
20 -->
21 <identity>
22 <dns value="localhost" />
23 </identity>
24 </endpoint>
25 <endpoint address="mex" binding="mexHttpBinding" contract="IMetadataExchange" />
26 </service>
27 </services>
28 </system.serviceModel>
29

```

- Esempio di un "app.config", autogenerato dall'IDE per un client WCF che si collega ad un webservice remoto esposto tramite AXIS (tecnologia Java) ed internamente implementato in Powerbuilder:

```

1  <?xml version="1.0" encoding="utf-8" ?>
2  <configuration>
3  <system.serviceModel>
4  <bindings>
5  <basicHttpBinding>
6  <binding name="n_gs_00_010_segnalazioni_s" closeTimeout="00:01:00"
7  openTimeout="00:01:00" receiveTimeout="00:10:00" sendTimeout="00:01:00"
8  allowCookies="false" bypassProxyOnLocal="false" hostNameComparisonMode="StrongWildcard"

```

```

9   maxBufferSize="65536" maxBufferPoolSize="524288" maxReceivedMessageSize="65536"
10  messageEncoding="Text" textEncoding="utf-8" transferMode="Buffered"
11  useDefaultWebProxy="true">
12  <readerQuotas maxDepth="32" maxStringContentLength="8192" maxArrayLength="16384"
13  maxBytesPerRead="4096" maxNameTableCharCount="16384" />
14  <security mode="None">
15  <transport clientCredentialType="None" proxyCredentialType="None"
16  realm="" />
17  <message clientCredentialType="UserName" algorithmSuite="Default" />
18  </security>
19  </binding>
20  </basicHttpBinding>
21  </bindings>
22  <client>
23  <endpoint address="http://patroclo:8080/ws/services/n_gs_00_010_segnalazioni_s"
24  binding="basicHttpBinding" bindingConfiguration="n_gs_00_010_segnalazioni_s"
25  contract="ServiceReferenceWsGs.n_gs_00_010_segnalazioni_s"
26  name="n_gs_00_010_segnalazioni_s" />
27  </client>
28  </system.serviceModel>
29  </configuration>
30

```

Note e soluzioni ad alcuni problemi

- When you're hosting WCF services within IIS 5 or 6 you can only use bindings that use the HTTP protocol, we can't use like the TCP or NamedPipes bindings etc.
- IIS 7 has WAS (Windows Process Activation Service) and that will allow to use other bindings.
- Hosting WCF - .svc Files in IIS7 (HTTP Error 404.3 - Not Found) =>
<http://thesherpaproject.com/2007/01/11/hosting-wcf-svc-files-in-iis7/>
- Se avete IIS 6 e non riuscite ad aprire il file .svc, provate le soluzioni proposte in questo articolo:
<http://msdn.microsoft.com/en-us/library/ms752252.aspx>

```

1  cd "C:\Windows\Microsoft.NET\Framework\v3.0\Windows Communication Foundation"
2  ServiceModelReg.exe -i
3  Microsoft(R) Windows Communication Foundation Installation Utility
4  [Microsoft (R) Windows (R) Communication Foundation, Version 3.0.4506.4037]
5  Copyright (c) Microsoft Corporation. All rights reserved.
6
7  Installing: Machine.config Section Groups and Handlers
8  Installing: System.Web Build Provider
9  Installing: System.Web Compilation Assemblies
10 Installing: HTTP Handlers
11 Installing: HTTP Modules
12 Installing: ListenerAdapter node for protocol net.tcp
13 Installing: Protocol node for protocol net.tcp
14 Installing: TransportConfiguration node for protocol net.tcp
15 Installing: ListenerAdapter node for protocol net.pipe
16 Installing: Protocol node for protocol net.pipe
17 Installing: TransportConfiguration node for protocol net.pipe
18 Installing: ListenerAdapter node for protocol net.msmq
19 Installing: Protocol node for protocol net.msmq
20 Installing: TransportConfiguration node for protocol net.msmq
21 Installing: ListenerAdapter node for protocol msmq.formatname
22 Installing: Protocol node for protocol msmq.formatname
23 Installing: TransportConfiguration node for protocol msmq.formatname
24 Installing: HTTP Modules (WAS)

```

WCF Test Client (WcfTestClient.exe)

Windows Communication Foundation (WCF) Test Client (WcfTestClient.exe) is a GUI tool that enables users to input test parameters, submit that input to the service, and view the response that the service sends back. It provides a seamless service testing experience when combined with WCF Service Host.

Esempi di utilizzo:

- Visual Studio 2008 Command Prompt => WcfTestClient.exe
- "C:\Program Files\Microsoft Visual Studio 9.0\Common7\IDE\WcfTestClient.exe"
 http://localhost/up/trunk/up_web_sln/up_ws/Services/ImportService.svc/mex
- "C:\Program Files\Microsoft Visual Studio 9.0\Common7\IDE\WcfTestClient.exe"
 http://localhost/up/trunk/up_web_sln/up_ws/Services/ImportService.svc
- "C:\Program Files\Microsoft Visual Studio 9.0\Common7\IDE\WcfTestClient.exe"
 http://localhost/up/trunk/up_web_sln/up_ws/Services/ImportService.svc?wsdl

Configuration Editor Tool (SvcConfigEditor.exe)

The Windows Communication Foundation (WCF) Service Configuration Editor (SvcConfigEditor.exe) allows administrators and developers to create and modify configuration settings for WCF services using a graphical user interface. With this tool, you can manage settings for WCF bindings, behaviors, services, and diagnostics without having to directly edit XML configuration files.

Esempi di utilizzo:

- Visual Studio 2008 Command Prompt => SvcConfigEditor.exe
- "C:\Program Files\Microsoft SDKs\Windows\v6.0A\bin\SvcConfigEditor.exe" => File > Open ...
Selezionare app.config oppure web.config del progetto.

Riferimenti

- [MSDN - WCF](#)
- [WCF Binding Comparison](#)
- [Codeproject WCF : 10 security related FAQ](#)