



Installing and Configuring .NET Agent Extensions for WCF

This document contains the contents of CA Wily Support Knowledge Base article #1617: https://support.wilytech.com/cgi-bin/wilytech.cfg/php/enduser/std_adp.php?p_faqid=1617.

This article covers the following topics:

- WCF background
- Installing .NET Agent extensions for WCF
- Configuring .NET Agent extensions for WCF
- Uninstalling the .NET Agent extension for WCF

Windows Communication Foundation (WCF) background

Windows Communication Foundation (WCF) is a distributed communication technology that is included in .NET Framework 3.0. WCF provides nine types of bindings to configure a WCF service:

- BasicHttpBinding
- WSHttpBinding
- WSDualHttpBinding
- WSFederationHttpBinding
- NetTcpBinding
- NetNamedPipeBinding
- NetMsmqBinding
- NetPeerTcpBinding
- MsmqIntegrationBinding

Web Services Manager 7.2 supports only these two bindings:

- BasicHttpBinding
- WSHttpBinding

Only WCF services that are built using these two bindings can be monitored by the Web Services Manager 7.2 .NET Agent extension for WCF.

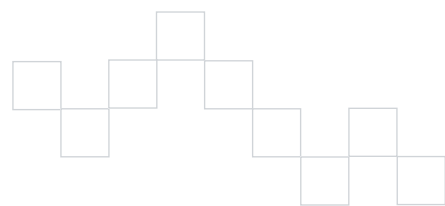
Web Services Manager 7.2 requires installation and configuration of an additional Agent extension to monitor WCF services and WCF clients calling WCF services.

Installing .NET Agent extensions for WCF

» **Important** Only .NET Framework 3.0 is supported.

Prerequisites

- .NET Framework 3.0 must be installed.
- Introscope .NET Agent version 7.2.1.0 or later must be installed and configured. See the *Introscope .NET Agent Guide*.
- If you upgraded your .NET Agent, you need to do additional configuration. See the section on **Reconfiguring after upgrading the Introscope .NET Agent version** in the *Web Services Manager Implementation Guide*.
- Web Services Manager 7.2 for .NET must be installed and deployed. See the section on **Installing and configuring .NET Agent extensions** in the *Web Services Manager Implementation Guide*.



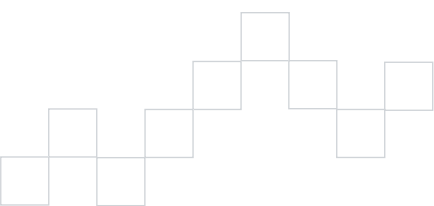
To install the .NET Agent extension for WCF

- 1 If needed, download Web Services Manager Agent extension installation archive file `WilyWebServicesManagerv7.2.0.0DotNet.zip` from the Software Download Site. Otherwise, find the location to which `WilyWebServicesManagerv7.2.0.0DotNet.zip` was downloaded. See the section on **Installing and configuring .NET Agent extensions** in the *Web Services Manager Implementation Guide*.
- 2 Use WinZip to extract all files in the `WilyWebServicesManagerv7.2.0.0Agent.windows.wcf.zip` installation archive file to `<Agent_Home>`, which contains the Agent configuration and is typically called `wily`.

The installation archive contains these files:

File	Description
<code>wilyregtool20.exe</code>	<p>Tool used to register the <code>WCFServices.Observer.dll</code> file.</p> <p>During the extraction process, the <code>wilyregtool20.exe</code> file is copied to the <code><Agent_Home>\bin</code> directory.</p>
<code>wily.WCFServicesAgent.ext.dll</code>	<p>Contains the .NET WCF-specific Web Services Manager Agent extension classes.</p> <p>During the extraction process, the <code>wily.WCFServicesAgent.ext.dll</code> file is copied to the <code><Agent_Home>\ext</code> directory.</p>
<code>WCFServices.Observer.dll</code>	<p>Contains the classes related to identifying .NET WCF-specific web services.</p> <p>During the extraction process, the <code>WCFServices.Observer.dll</code> file is copied to the <code><Agent_Home>\bin</code> directory.</p>

The .NET Agent extension for WCF is installed.



Configuring .NET Agent extensions for WCF

Several files need to be configured before you can monitor and view WCF-specific web services using the .NET Agent extension for WCF. Configuring the .NET Agent extension for WCF includes these high-level steps:

- Step 1** Registering the WCFServices.dll
- Step 2** Defining WebServicesObserver as the endpoint behavior
- Step 3** Adding WebServicesObserver as the behavior extension
- Step 4** Adding WebServicesObserver as a common behavior
- Step 5** Adding the behavior configuration to the WCF services endpoints
- Step 6** Modifying the webservices.pbd file

Registering the WCFServices.dll

Use `wilyregtool20.exe` to register the `WCFServices.Observer.dll` file.

- 1 Open a command prompt.
- 2 Go to `<Agent_Home>\bin`.
- 3 Use this `wilyregtool20.exe` command to register the `WCFServices.Observer.dll`:

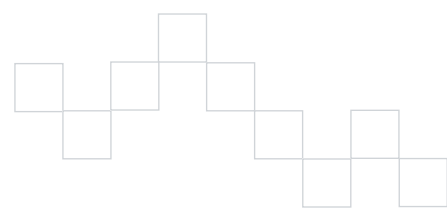
```
wilyregtool20.exe "WCFServices.Observer.dll"
```

Defining WebServicesObserver as the endpoint behavior

- 1 Back up to a secure location the `machine.config` file, which is located under .NET Framework version 2.0. Typically `machine.config` is found here:

```
%WINDIR%\Microsoft.NET\Framework\v2.0.50727\CONFIG
```

- 2 Open the `machine.config` file.
- 3 Check for `<system.serviceModel> | <behaviors> | <endpointBehaviors>`.
 - » **Note** CA Wily uses the pipe character (|) to denote node or resource levels. In this case, `<endpointBehaviors>` to the right of `<behaviors>` indicates that the `<endpointBehaviors>` node is located underneath the `<behaviors>` node.
- 4 Add the appropriate code below depending on the node names you found in step 3 above.



This code is required to monitor the WCF clients and WCF services.

If you found that	Then go to this sub-step below
<code><endpointBehaviors></code> is present	step a on page 5
<code><endpointBehaviors></code> is not present	step b on page 5
<code><behaviors></code> is not present	step c on page 5
<code><system.serviceModel></code> is not present	step d on page 6

a If `<endpointBehaviors>` is present, add this code under

```
<endpointBehaviors>:
<behavior name="WebServicesObserverBehavior">
<WebServicesObserver />
</behavior>
```

b If `<endpointBehaviors>` is not present, add this code under `<behaviors>`:

```
<endpointBehaviors>
  <behavior name="WebServicesObserverBehavior">
    <WebServicesObserver />
  </behavior>
</endpointBehaviors>
```

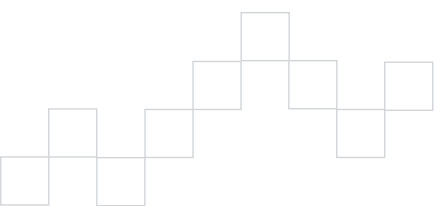
c If `<behaviors>` is not present, add this code under

```
<system.serviceModel>:

<behaviors>
  <endpointBehaviors>
    <behavior name="WebServicesObserverBehavior">
      <WebServicesObserver />
    </behavior>
  </endpointBehaviors>
</behaviors>
```

When you are finished adding the code, the endpoint behaviors code in the `machine.config` file looks like this:

```
<behaviors>
  <endpointBehaviors>
    <behavior name="WebServicesObserverBehavior">
      <WebServicesObserver />
    </behavior>
```



```
.....other endpoint behaviors.....
```

```
    </endpointBehaviors>
```

```
  </behaviors>
```

d If `<system.serviceModel>` is not present, add this code at end of the `machine.config` file just above `</configuration>`:

```
<system.serviceModel>
```

```
  <behaviors>
```

```
    <endpointBehaviors>
```

```
      <behavior name="WebServicesObserverBehavior">
```

```
        <WebServicesObserver />
```

```
      </behavior>
```

```
    </endpointBehaviors>
```

```
  </behaviors>
```

```
</system.serviceModel>
```

Adding WebServicesObserver as the behavior extension

1 In the `machine.config` file, find `<system.serviceModel>` | `<extensions>` | `<behaviorExtensions>`.

2 Add this code under `<behaviorExtensions>`:

```
<add name="WebServicesObserver"
```

```
  type="com.wily.powerpack.webservices.extension.MessageInspectorE
```

```
xtensionElement, WCFServices.Observer, Version=7.2.0.0,
```

```
  Culture=neutral, PublicKeyToken=99330fac7173009e" />
```

When you are finished adding the code, the behavior extension code in the `machine.config` file looks like this:

```
<extensions>
```

```
  <behaviorExtensions>
```

```
    <add name="WebServicesObserver"
```

```
  type="com.wily.powerpack.webservices.extension.MessageInspectorE
```

```
xtensionElement, WCFServices.Observer, Version=7.2.0.0,
```

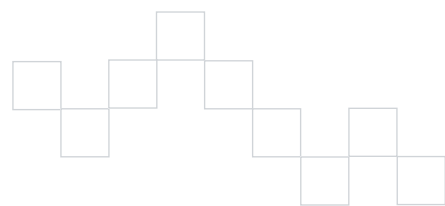
```
  Culture=neutral, PublicKeyToken=99330fac7173009e" />
```

```
.....other behavior extensions...
```

```
  </behaviorExtensions>
```

```
.....other extensions...
```

```
</extensions>
```



Adding WebServicesObserver as a common behavior

WebServicesObserver is automatically applied to all the WCF clients running on the system.

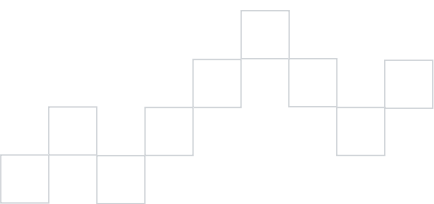
- 1 In the `machine.config` file, find `<system.serviceModel>` | `<commonBehaviors>` | `<endpointBehaviors>`.
- 2 Add `<WebServicesObserver />` under `<endpointBehaviors>`.

When you are finished adding the code, the common behavior code in the `machine.config` file looks like this:

```
<commonBehaviors>
  <endpointBehaviors>
    <WebServicesObserver />
  </endpointBehaviors>
</commonBehaviors>
```

Now the Web Services Manager WCF-related code in the `machine.config` file looks like this:

```
<system.serviceModel>
  <behaviors>
    <endpointBehaviors>
      <behavior name="WebServicesObserverBehavior">
        <WebServicesObserver />
      </behavior>
    </endpointBehaviors>
  </behaviors>
  <extensions>
    <behaviorExtensions>
      <add name="WebServicesObserver"
type="com.wily.powerpack.webservices.extension.MessageInspectorE
xtensionElement, WCFServices.Observer, Version=7.2.0.0,
Culture=neutral, PublicKeyToken=99330fac7173009e" />
```



```

        </behaviorExtensions>
    </extensions>
    <commonBehaviors>
        <endpointBehaviors>
            <WebServicesObserver />
        </endpointBehaviors>
    </commonBehaviors>
</system.serviceModel>

```

Adding the behavior configuration to the WCF services endpoints

- 1 Open the `web.config` file of the WCF service that needs to be monitored.
- 2 Add this code to the endpoints that implement `wsHttpBinding` and `basicHttpBinding`:

```
behaviorConfiguration="WebServicesObserverBehavior"
```

When you are finished adding the code, the `web.config` file looks like this:

- For `basicHttpBinding`

```

<endpoint address="" binding="basicHttpBinding"
contract=" Com.Wily.ServiceModel.Samples.ICalculator"
behaviorConfiguration="WebServicesObserverBehavior"/>

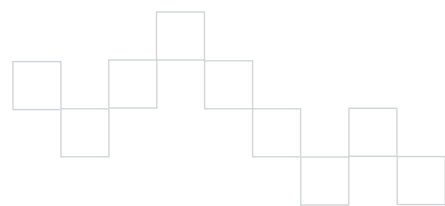
```

- For `wsHttpBinding`

```

<endpoint address="" binding="wsHttpBinding"
bindingConfiguration="Binding1" contract="
Com.Wily.ServiceModel.Samples.ICalculator"
behaviorConfiguration="WebServicesObserverBehavior"/>

```



- » **Note** Note: If `behaviorConfiguration` has already been defined in the endpoints, add `<WebServicesObserver/>` under the behavior name, as shown in bold in the code sample below.

For example if `behaviorConfiguration` is defined as follows

`behaviorConfiguration="TestBehavior"`, then the configuration looks like this:

```
<endpointBehaviors>
  <behavior name="TestBehavior" >
    .....other behavior....
    <WebServicesObserver/>
  </behavior>
</endpointBehaviors>
```

Modifying the `webservices.pbd` file

- 1 Open the `webservices.pbd` located in the `<Agent_Home>` directory.
- 2 Search for `WCF Services Tracing` and follow the instructions written there.

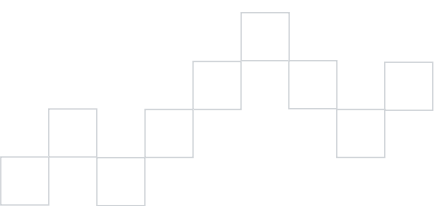
If the service contract interface is for example,

`Com.Wily.ServiceModel.Samples.ICalculator`, then the `webservices.pbd` code should look like this:

```
SetFlag: WCFServerTracing
TurnOn: WCFServerTracing
```

```
IdentifyInheritedAs: Com.Wily.ServiceModel.Samples.ICalculator
WCFServerTracing
```

```
TraceAllMethodsIfFlagged: WCFServerTracing
WebServicesProducerAggregatedTracer
"WebServices|Server|{namespace}"
TraceAllMethodsIfFlagged: WCFServerTracing
WebServicesProducerTracer
"WebServices|Server|{namespace}|{method}"
TraceAllMethodsIfFlagged: WCFServerTracing
WebServicesProducerErrorTracer
"WebServices|Server|{namespace}:Errors Per Interval"
TraceAllMethodsIfFlagged: WCFServerTracing
WebServicesProducerErrorTracer
"WebServices|Server|{namespace}|{method}:Errors Per Interval"
TraceAllMethodsIfFlagged: WCFServerTracing
WebServicesCrossProcessTracer "WebServices|Server"
```



3 Search for WCF Client Tracing and follow the instructions written there.

If the WCF client proxy class is for example

Com.Wily.ServiceModel.Clients.CalculatorClient the webservices.pbd code should look like this:

```
SetTracerClassMapping: WebServicesWCFClientTracer
    com.wily.introscope.agent.trace.BlamePointTracer
    com.wily.introscope.probebuilder.validate.ResourceNameValidator
SetTracerParameter: WebServicesWCFClientTracer nameformatter
    com.wily.powerpack.webservices.extension.agent.trace.nameformatter.dotnet.WCFClientNameFormatter

SetTracerClassMapping: WebServicesWCFBackendTracer
    com.wily.introscope.agent.trace.BackendTracer
    com.wily.introscope.probebuilder.validate.ResourceNameValidator
SetTracerParameter: WebServicesWCFBackendTracer nameformatter
    com.wily.powerpack.webservices.extension.agent.trace.nameformatter.dotnet.WCFClientNameFormatter
SetTracerOrdering: WebServicesWCFBackendTracer -1000

SetTracerClassMapping: WebServicesWCFClientErrorTracer
    com.wily.powerpack.webservices.extension.agent.trace.fault.dotnet.ErrorReportingTracer
    com.wily.introscope.probebuilder.validate.MetricNameValidator
SetTracerParameter: WebServicesWCFClientErrorTracer nameformatter
    com.wily.powerpack.webservices.extension.agent.trace.nameformatter.dotnet.WCFClientNameFormatter

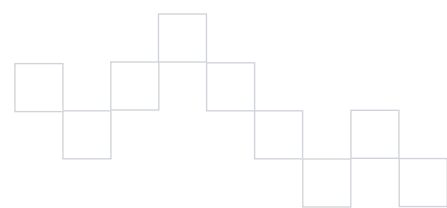
SetFlag: WCFClientTracing
TurnOn: WCFClientTracing

IdentifyClassAs: Com.Wily.ServiceModel.Clients.CalculatorClient
    WCFClientTracing

TraceAllMethodsIfFlagged: WCFClientTracing WebServicesWCFBackendTracer
    "WebService at {namespace}"
TraceAllMethodsIfFlagged: WCFClientTracing WebServicesWCFClientTracer
    "WebServices|Client|{namespace}"
TraceAllMethodsIfFlagged: WCFClientTracing WebServicesWCFClientTracer
    "WebServices|Client|{namespace}|{method}"
TraceAllMethodsIfFlagged: WCFClientTracing WebServicesWCFClientErrorTracer
    "WebServices|Client|{namespace}:Errors Per Interval"
TraceAllMethodsIfFlagged: WCFClientTracing WebServicesWCFClientErrorTracer
    "WebServices|Client|{namespace}|{method}:Errors Per Interval"
TraceAllMethodsIfFlagged: WCFClientTracing WebServicesCrossProcessTracer
    "WebServices|Client"
```

4 Restart the IIS application server.

The Web Services Manager .NET Agent extension is configured for WCF.



Uninstalling the .NET Agent extension for WCF

To uninstall the .NET Agent extension for WCF

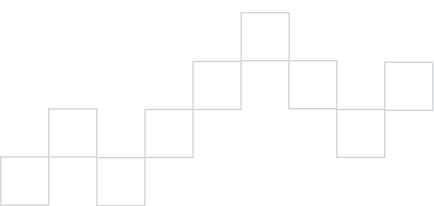
- 1 Stop the IIS application server.
- 2 Use `wilyregtool20.exe` to unregister the `WCFServices.ObsERVER.dll` file.
 - a Open a command prompt.
 - b Go to `<Agent_Home>\bin`.
 - c Use this `wilyregtool20.exe` command to register the `WCFServices.ObsERVER.dll`.

```
Wilyregtool20.exe -u "WCFServices.ObsERVER.dll"
```
- 3 Remove code from the `machine.config` file.
 - a Locate and open the `machine.config` file.
 - » **Note** The `machine.config` file is typically located in the `C:\WINDOWS\Microsoft.NET\Framework\vx.x\CONFIG` directory, where `vx.x` is the .NET version. For example, the directory for .NET 1.1 might be `v1.1.4322` and the directory for .NET 2.0 might be `v2.0.50727`.
 - b Remove this code under `<endpointBehaviors>`:

```
<behavior name="WebServicesObserverBehavior">
<WebServicesObserver />
```
 - c Remove `WebServicesObserver` as the behavior extension by removing this code under `<behaviorExtensions>`:

```
<add name="WebServicesObserver"
type="com.wily.powerpack.webservices.extension.MessageInspectorE
xtensionElement, WCFServices.ObsERVER, Version=7.2.0.0,
Culture=neutral, PublicKeyToken=99330fac7173009e" />
```
- 4 Remove `WebServicesObserver` as a common behavior by removing `<WebServicesObserver />` under `<endpointBehaviors>`.

`WebServicesObserver` is automatically removed from all the WCF clients running on the system.



- 5 Remove the behavior configuration from the WCF services endpoints.
 - a Open the web.config file of the WCF service that is being monitored.
 - b Remove this code from the endpoints that implement `wsHttpBinding` and `basicHttpBindings`:

```
behaviorConfiguration="WebServicesObserverBehavior"
```

When you are finished removing the code, the web.config file looks like this:

- For `basicHttpBinding`

```
<endpoint address="" binding="basicHttpBinding"
contract=" Com.Wily.ServiceModel.Samples.ICalculator"/>
```

- For `wsHttpBinding`

```
<endpoint address=""
            binding="wsHttpBinding"
            bindingConfiguration="Binding1"
            contract=" Com.Wily.ServiceModel.Samples.ICalculator"/>
```

- 6 Unmodify the `webservices.pbd` file located in the `<Agent_Home>` directory.
 - a Search for `WCF Services Tracing` and reverse the instructions written there.
 - b Search for `WCF Client Tracing` and reverse the instructions written there.
- 7 Delete the .NET Agent extension for WCF files.
 - a Delete `WCFServices.Observer.dll` from the `<Agent_Home>\bin` directory.
 - b Delete `wily.WCFServicesAgent.ext.dll` from the `<Agent_Home>\ext` directory.
 - c Delete `wilyregtool20.exe` from the `<Agent_Home>\bin` directory.
- 8 Start the IIS application server.

The .NET Agent extension is uninstalled.

