

# Usando Windows Management Instrumentation para diagnóstico

30/03/2017 • 8 minutos para o fim da leitura • ●●

## Neste artigo

[Habilitando o WMI](#)

[Acessando dados WMI](#)

[Segurança](#)

[Concessão de Permissões de Registro WCF WMI para usuários adicionais](#)

[Acessando instâncias remotas de objeto WMI](#)

A Windows Communication Foundation (WCF) expõe os dados de inspeção de um serviço em tempo de execução através de um provedor WCF Windows Management Instrumentation (WMI).

## Habilitando o WMI

O WMI é a implementação da Microsoft do padrão WBEM (Web-Based Enterprise Management, gerenciamento de empresas baseado na Web). Para obter mais informações sobre o WMI SDK, consulte [A Instrumentação de Gerenciamento do Windows](#). O WBEM é um padrão do setor para como as aplicações expõem a instrumentação de gerenciamento a ferramentas de gestão externa.

Um provedor WMI é um componente que expõe a instrumentação em tempo de execução através de uma interface compatível com WBEM. Consiste em um conjunto de objetos WMI que têm pares de atributos/valor. Pares podem ser de vários tipos simples. As ferramentas de gerenciamento podem se conectar aos serviços através da interface em tempo de execução. O WCF expõe atributos de serviços como endereços, vinculações, comportamentos e ouvintes.

O provedor WMI incorporado pode ser ativado no arquivo de configuração do aplicativo. Isso é feito `wmiProviderEnabled` através do atributo do `<>de diagnóstico` na seção `<system.serviceModel>`, conforme mostrado na configuração da amostra a seguir.

XML	 Copiar
<pre>&lt;system.serviceModel&gt; ...   &lt;diagnostics wmiProviderEnabled="true" /&gt; ... &lt;/system.serviceModel&gt;</pre>	

Esta entrada de configuração expõe uma interface WMI. Os aplicativos de gerenciamento agora podem se conectar através desta interface e acessar a instrumentação de gerenciamento do aplicativo.

# Acessando dados WMI

Os dados WMI podem ser acessados de muitas maneiras diferentes. A Microsoft fornece APIs WMI para scripts, aplicativos Visual Basic, aplicativos C++ e o .NET Framework. Para obter mais informações, consulte [Usando WMI](#).

## ⊗ Cuidado

Se você usar os métodos fornecidos pelo .NET Framework para acessar programaticamente os dados do WMI, você deve estar ciente de que tais métodos podem lançar exceções quando a conexão for estabelecida. A conexão não é estabelecida **ManagementObject** durante a construção da instância, mas na primeira solicitação envolvendo troca de dados real. Portanto, você deve `try..catch` usar um bloco para capturar as possíveis exceções.

Você pode alterar o nível de registro de rastreamento `System.ServiceModel` e mensagens, bem como opções de registro de mensagens para a fonte de rastreamento no WMI. Isso pode ser feito acessando a instância [AppDomainInfo](#), `LogMessagesAtTransportLevel` que `LogMalformedMessages` expõe `TraceLevel` essas propriedades booleanas: `LogMessagesAtServiceLevel`, , e . Portanto, se você configurar um ouvinte de rastreamento para registro `false` de mensagens, mas `true` definir essas opções na configuração, você poderá alterá-las posteriormente para quando o aplicativo estiver sendo executado. Isso permitirá efetivamente o registro de mensagens em tempo de execução. Da mesma forma, se você habilitar o registro de mensagens em seu arquivo de configuração, você poderá desativá-lo em tempo de execução usando o WMI.

Você deve estar ciente de que se nenhum registro `System.ServiceModel` de mensagem rastrear ouvintes para registro de mensagens ou nenhum rastreamento de rastreamento for especificado no arquivo de configuração, nenhuma de suas alterações será tomada em vigor, mesmo que as alterações sejam aceitas pelo WMI. Para obter mais informações sobre a configuração adequada dos respectivos ouvintes, consulte [Configurando o registro de mensagens](#) e [configurando o rastreamento](#). O nível de rastreamento de todas as outras fontes de rastreamento especificadas pela configuração é eficaz quando o aplicativo é iniciado e não pode ser alterado.

O WCF `GetOperationCounterInstanceName` expõe um método de scripting. Este método retorna um nome de instância de contador de desempenho se você fornecê-lo com um nome de operação. No entanto, ele não valida sua entrada. Portanto, se você fornecer um nome de operação incorreto, um nome de contador incorreto será devolvido.

A `OutgoingChannel` propriedade `Service` da instância não conta canais abertos por um serviço para se conectar a outro serviço, `Service` caso o cliente WCF ao serviço de destino não seja criado dentro do método.

**Cuidado** O WMI só [TimeSpan](#) suporta um valor de até 3 pontos decimais. Por exemplo, se o seu serviço [MaxValue](#) define uma de suas propriedades para , seu valor é truncado após 3 pontos decimais

quando visualizado através de WMI.

## Segurança

Como o provedor WCF WMI permite a descoberta de serviços em um ambiente, você deve ter extrema cautela para conceder acesso a ele. Se você relaxar o acesso padrão somente ao administrador, poderá permitir que partes menos confiáveis acessem dados confidenciais em seu ambiente. Especificamente, se você afrouxar permissões em acesso remoto WMI, ataques de inundação podem ocorrer. Se um processo for inundado por solicitações excessivas de WMI, seu desempenho pode ser degradado.

Além disso, se você relaxar as permissões de acesso para o arquivo MOF, as partes menos confiáveis podem manipular o comportamento do WMI e alterar os objetos que estão carregados no esquema WMI. Por exemplo, os campos podem ser removidos de forma que os dados críticos sejam ocultados do administrador ou que campos que não preencham ou causem exceções sejam adicionados ao arquivo.

Por padrão, o provedor WCF WMI concede permissão de "executar método", "gravação de provedor" e "habilitar conta" para administrador e "habilitar conta" para ASP.NET, Serviço Local e Serviço de Rede. Em particular, em plataformas não-Windows Vista, a conta ASP.NET leu acesso ao namespace WMI ServiceModel. Se você não quiser conceder esses privilégios a um determinado grupo de usuários, você deve desativar o provedor WMI (ele está desativado por padrão) ou desativar o acesso para o grupo de usuários específico.

Além disso, quando você tenta ativar o WMI através da configuração, o WMI pode não ser habilitado devido ao privilégio insuficiente do usuário. No entanto, nenhum evento é escrito no registro do evento para registrar essa falha.

Para modificar os níveis de privilégio do usuário, use as seguintes etapas.

1. Clique em Iniciar e, em seguida, Executar e digitar **compmgmt.msc**.
2. Clique com o botão direito do mouse em **Serviços e controles de aplicativo/WMI** para selecionar **Propriedades**.
3. Selecione a guia **de segurança** e navegue até o namespace **Root/ServiceModel**. Clique no botão **Segurança**.
4. Selecione o grupo ou usuário específico que deseja controlar o acesso e use a caixa de seleção **Permitir** ou **Negar** para configurar permissões.

## Concessão de Permissões de Registro WCF WMI para usuários adicionais

O WCF expõe os dados de gerenciamento ao WMI. Ele faz isso hospedando um provedor WMI em processo, às vezes chamado de "provedor dissociado". Para que os dados de gerenciamento sejam

expostos, a conta que registra este provedor deve ter as permissões apropriadas. No Windows, apenas um pequeno conjunto de contas privilegiadas pode registrar provedores dissociados por padrão. Isso é um problema porque os usuários geralmente querem expor dados WMI de um serviço WCF em execução em uma conta que não está no conjunto padrão.

Para fornecer esse acesso, um administrador deve conceder as seguintes permissões à conta adicional na seguinte ordem:

1. Permissão para acesso ao WCF WMI Namespace.
2. Permissão para registrar o Provedor WMI Desacoplado WCF.

## Para conceder permissão de acesso ao namespace do WMI

1. Execute o script do PowerShell a seguir.

PowerShell

 Copiar

```
write-host ""
write-host "Granting Access to root/servicemodel WMI namespace to built in users
group"
write-host ""

# Create the binary representation of the permissions to grant in SDDL
$newPermissions = "O:BAG:BAD:P(A;CI;CCDCLCSWRPWPWCWD;;;BA)(A;CI;CC;;;NS)
(A;CI;CC;;;LS)(A;CI;CC;;;BU)"
$converter = new-object system.management.ManagementClass
Win32_SecurityDescriptorHelper
$binarySD = $converter.SDDLToBinarySD($newPermissions)
$convertedPermissions = , $binarySD.BinarySD

# Get the object used to set the permissions
$security = gwmi -namespace root/servicemodel -class __SystemSecurity

# Get and output the current settings
$binarySD = @($null)
$result = $security.PsBase.InvokeMethod("GetSD", $binarySD)

$outsdll = $converter.BinarySDToSDDL($binarySD[0])
write-host "Previous ACL: " $outsdll.SDDL

# Change the Access Control List (ACL) using SDDL
$result = $security.PsBase.InvokeMethod("SetSD", $convertedPermissions)

# Get and output the current settings
$binarySD = @($null)
$result = $security.PsBase.InvokeMethod("GetSD", $binarySD)

$outsdll = $converter.BinarySDToSDDL($binarySD[0])
write-host "New ACL:      " $outsdll.SDDL
write-host ""
```

Este script PowerShell usa o SDDL (Security Descriptor Definition Language, linguagem de definição de descritor de segurança) para conceder ao grupo de usuários incorporados acesso ao espaço de nome WMI "root/servicemodel". Ele especifica as seguintes ACLs:

- Administrador embutido (BA) - Já tinha acesso.
- Serviço de Rede (NS) - Já tinha acesso.
- Sistema Local (LS) - Já tinha acesso.
- Usuários Incorporados - O grupo para conceder acesso.


## Para conceder acesso ao registro do provedor

1. Execute o script do PowerShell a seguir.

PowerShell	 Copiar
<pre>write-host "" write-host "Granting WCF provider registration access to built in users group" write-host "" # Set security on ServiceModel provider \$provider = get-WmiObject -namespace "root\servicemodel" __Win32Provider  write-host "Previous ACL: "\$provider.SecurityDescriptor \$result = \$provider.SecurityDescriptor = "O:BUG:BUD:(A;;0x1;;;BA)(A;;0x1;;;NS) (A;;0x1;;;LS)(A;;0x1;;;BU)"  # Commit the changes and display it to the console \$result = \$provider.Put() write-host "New ACL:      "\$provider.SecurityDescriptor write-host ""</pre>	

## Concessão de acesso a usuários ou grupos arbitrários

O exemplo nesta seção concede privilégios de registro do Provedor WMI a todos os usuários locais. Se você quiser conceder acesso a um usuário ou grupo que não esteja incorporado, então você deve obter o SID (Security Identifier, identificador de segurança) desse usuário ou grupo. Não há uma maneira simples de obter o SID para um usuário arbitrário. Um método é fazer logon como usuário desejado e, em seguida, emitir o seguinte comando shell.

Console	 Copiar
<pre>Whoami /user</pre>	

Isso fornece o SID do usuário atual, mas este método não pode ser usado para obter o SID em qualquer usuário arbitrário. Outro método para obter o SID é usar a ferramenta [getsid.exe](#) das

Ferramentas de Kit de Recursos do Windows 2000 para tarefas administrativas. Esta ferramenta compara o SID de dois usuários (local ou domínio), e como efeito colateral imprime os dois SIDs para a linha de comando. Para obter mais informações, consulte [SIDs bem conhecidos](#).

## Acessando instâncias remotas de objeto WMI

Se você precisar acessar instâncias WCF WMI em uma máquina remota, você deve habilitar a privacidade do pacote nas ferramentas que você usa para acesso. A seção a seguir descreve como alcançá-los usando o WMI CIM Studio, O Tester de Instrumentação de Gerenciamento do Windows, bem como o .NET SDK 2.0.

### WMI CIM Studio

Se você instalou ferramentas administrativas WMI, você pode usar o WMI CIM Studio para acessar instâncias WMI. As ferramentas estão na seguinte pasta:

`%windir%\Arquivos do programa\WMI Ferramentas\`

1. No **Conecte-se ao namespace**: janela, digite `root\ServiceModel` e clique em **OK**.
2. Na janela De login do **WMI CIM Studio**, clique no botão **Opções >>** para expandir a janela. Selecione **privacidade do pacote** para o nível de **autenticação** e clique em **OK**.


## Testador de instrumentação de gerenciamento de Windows

Esta ferramenta é instalada pelo Windows. Para executá-lo, inicie um console de comando digitando `cmd.exe` na caixa de diálogo **Iniciar/Executar** e clique em **OK**. Em seguida, digite `wbemtest.exe` na janela de comando. A ferramenta Tester de Instrumentação de Gerenciamento do Windows é então lançada.

1. Clique no botão **Conectar** no canto superior direito da janela.
2. Na nova janela, digite `root\ServiceModel` para o campo **Namespace** e selecione **Privacidade de pacote** para nível de autenticação. Clique em **Conectar**.

## Usando código gerenciado

Você também pode acessar instâncias remotas do WMI [System.Management](#) programaticamente usando classes fornecidas pelo namespace. A amostra de código a seguir demonstra como fazer isso.

C#	 Copiar
<pre>String wcfNamespace = \$"{this.serviceMachineName}\Root\ServiceModel";  ConnectionOptions connection = new ConnectionOptions();</pre>	

```
connection.Authentication = AuthenticationLevel.PacketPrivacy;  
ManagementScope scope = new ManagementScope(this.wcfNamespace, connection);
```

---

Esta página é útil?

 Yes  No

---