

Test WMI Connectivity for Connector Service with WBEMTest

Contents

[Introduction](#)

[Test WMI Connectivity](#)

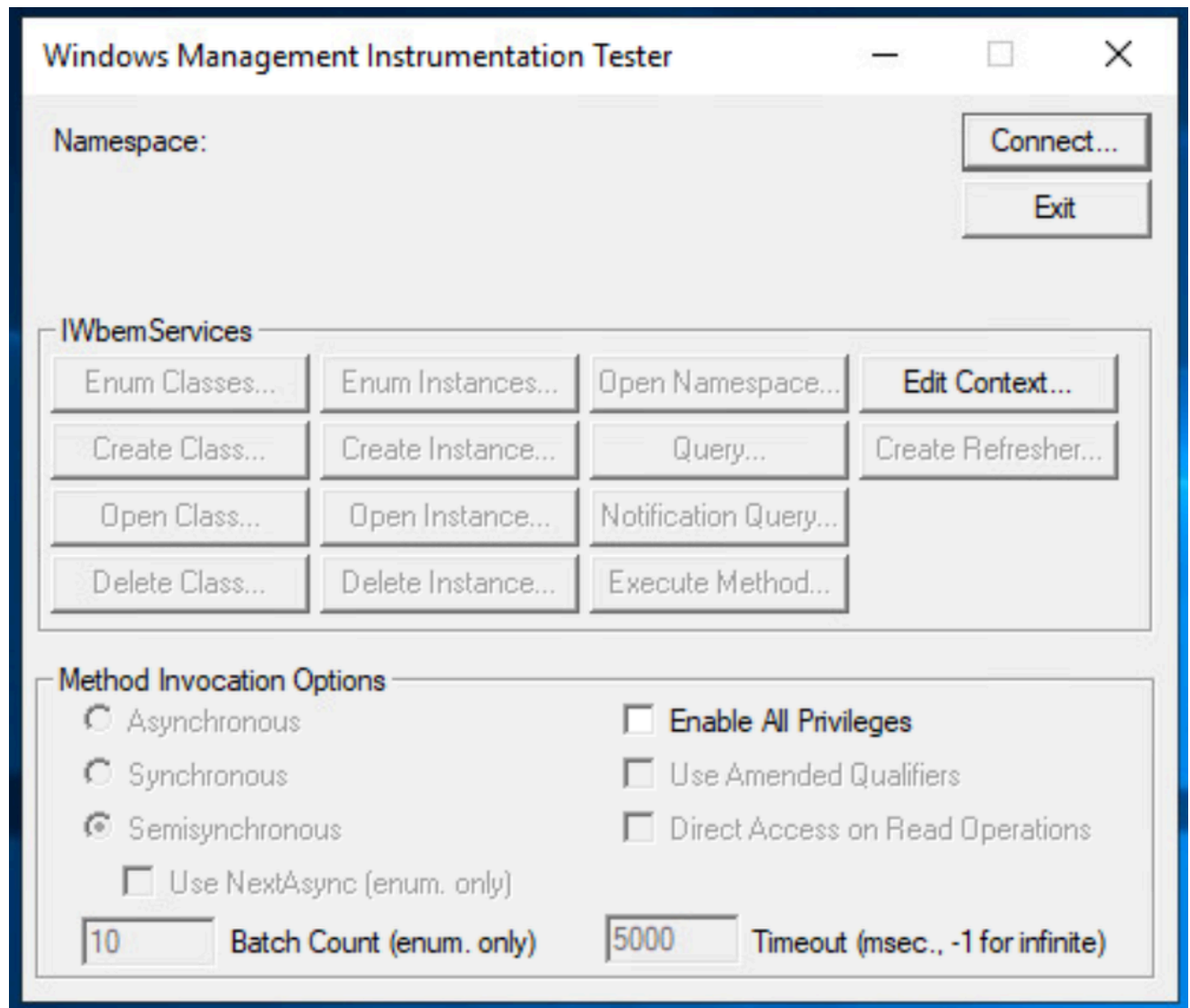
Introduction

This document describes how to manually test WMI connectivity for Connector Service with WBEMTest.

Test WMI Connectivity

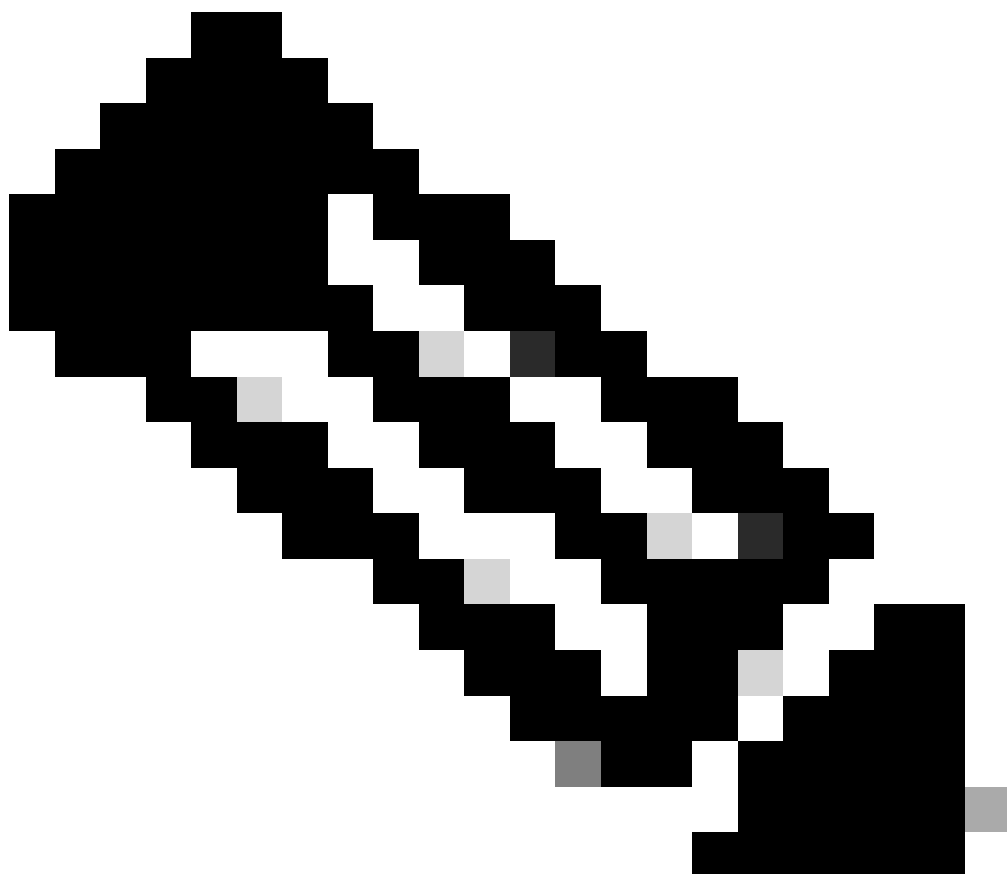
The Connector Service uses WMI in order to connect to your DCs and collect events. If the permissions for these connections are not set, you see "Access Denied" errors in your dashboard. This test must be run on the computer running the connector. These are the steps to manually test WMI connectivity:

1. Click **Start > Run > WBEMTest > OK**. This screen appears:



24075613964308

2. Click **Connect**.
3. Pick a DC which is experiencing the issue or which support has instructed you to run the test at. This must be a different machine than the one you are running this test on.
4. Type the IP you are trying to connect to in this format: \\192.168.10.1\root\cimv2. Do NOT use the hostname.



Note: Run the WBEMTest on the same machine as the connector is installed on and in this step, you are connecting to a remote DC.

-
5. Under credentials, use the OpenDNS_Connector user as well as the password.

Connect

Namespace

\\10.122.170.2\root\cimv2

Connect

Cancel

Connection:

Using: IwbemLocator (Namespaces)

Returning: IwbemServices

Completion: Synchronous

Credentials

User: OpenDNS_Connector

Password: xxxxxxxxxx

Authority:

Locale

How to interpret empty password

☒ NULL

☐ Blank

Impersonation level

☐ Identify

☒ Impersonate

☐ Delegate

Authentication level

☐ None

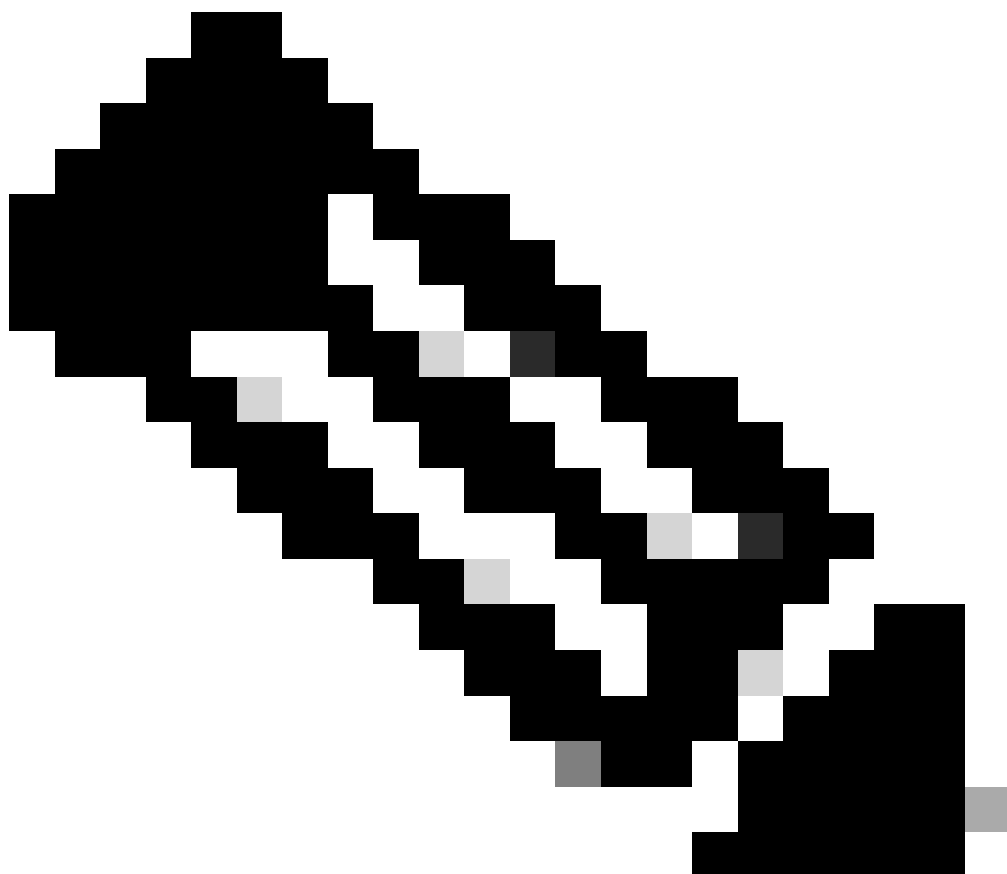
☒ Packet

☐ Connection

☐ Packet integrity

☐ Call

☐ Packet privacy



Note: To run this test directly on a DC, use "root\cimv2" as the Namespace and blank Username and Password fields.

6. Click **Connect**.

Windows Management Instrumentation Tester

Namespace:
root\cimv2

Connect...
Exit

IWbemServices

Enum Classes...	Enum Instances...	Open Namespace...	Edit Context...
Create Class...	Create Instance...	Query...	Create Refresher...
Open Class...	Open Instance...	Notification Query...	
Delete Class...	Delete Instance...	Execute Method...	

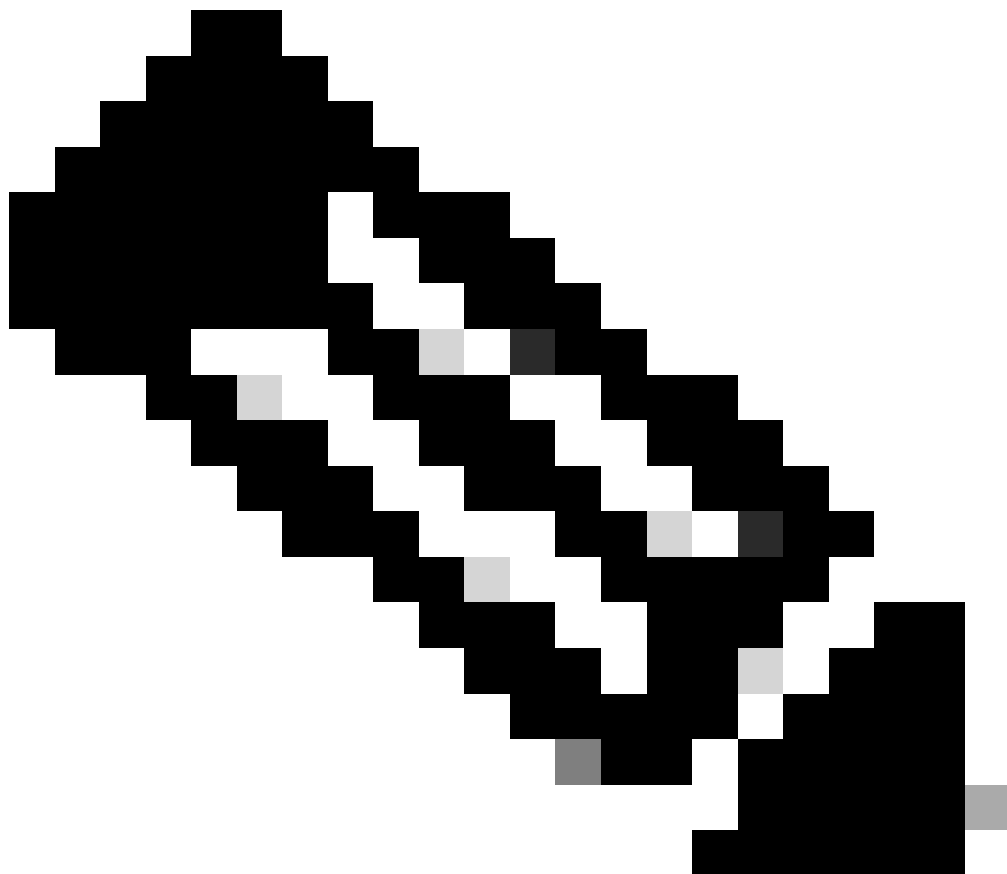
Method Invocation Options

☐ Asynchronous
☐ Synchronous
☒ Semisynchronous
☐ Use NextAsync (enum. only)

☐ Enable All Privileges
☐ Use Amended Qualifiers
☐ Direct Access on Read Operations

10 Batch Count (enum. only) 5000 Timeout (msec., -1 for infinite)

24075613972756



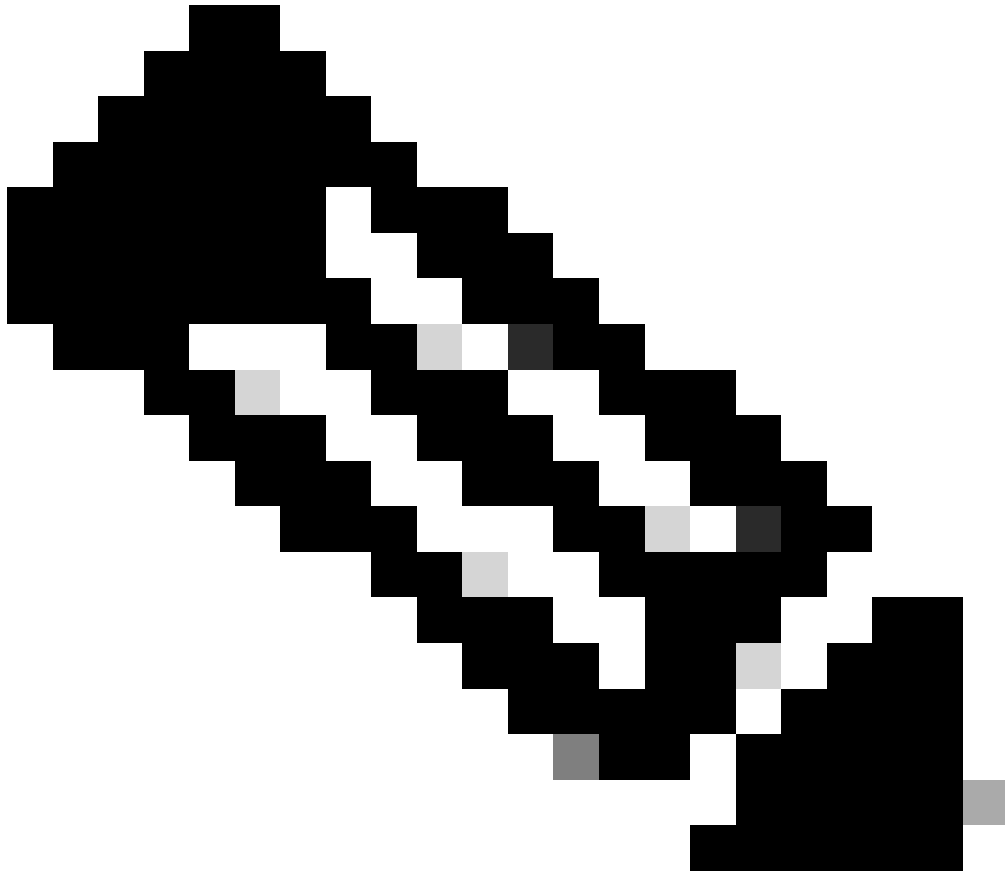
Note: Note: If you see an Access Denied error, check the WMI and DCOM permissions. DCOM requires a reboot if changed. See our Access Denied article [here](#).

7. Click **Notification Query**.

8. Paste this content inside the white text box:

```
SELECT * FROM __InstanceCreationEvent WHERE TargetInstance ISA 'Win32_NTLogEvent' and TargetInstan
```

9. Click **Apply**.



Note: Note: If you see an Access Denied error, ensure the Event Log Readers group can access the event logs with our help article [here](#).

-
10. If everything is working properly, you see objects, as shown in this image. Otherwise, you are either missing WMI permissions or having connection issues to the remote server.

Query Result

WQL: SELECT * FROM __InstanceCreationEvent WHERE TargetInstance IS

Close

42 objects

max. batch: 10

Operation in progress...

__InstanceCreationEvent=<no key>
__InstanceCreationEvent=<no key>
__InstanceCreationEvent=<no key>
__InstanceCreationEvent=<no key>
__InstanceCreationEvent=<no key>
__InstanceCreationEvent=<no key>
__InstanceCreationEvent=<no key>
__InstanceCreationEvent=<no key>
__InstanceCreationEvent=<no key>
__InstanceCreationEvent=<no key>
__InstanceCreationEvent=<no key>
__InstanceCreationEvent=<no key>

<

>

Add

Delete

24075601961620