

Installing Microsoft .NET tier collectors

This section includes the following topics:

- Pre-installation tasks
- Installing a Microsoft .NET Tier instance
- Post-installation action items

Pre-installation tasks

Most of the screens for this installation require you to input details regarding your information systems. Before performing this installation, compile a list of all required user inputs based on the screen names and tables below. Each table (in the sections below) lists the options per screen.

Before installing a Microsoft .NET Tier instance, verify that you have performed the following tasks:



The second and third tasks must be performed on the target server.

- Use the Precise Framework Installer to install framework components
- Create a Microsoft .NET login name with system administrator privileges and a password
- Close the Services window in the Control Panel.

Installing a Microsoft .NET Tier instance

This section describes the procedures that are needed to install a .NET Tier instance. You have the option of adding a new instance, as well as adding a new application. For information on editing or removing an existing instance, see [Editing or removing tier instances or clusters](#).

Adding a .NET Tier



Before adding a Tier instance, create a list of all required inputs based on the screen information below.

To add a Tier instance

1. From StartPoint, click the AdminPoint tab.
2. Click the Setup tab.
3. From the drop-down menu, select **Instances & Clusters**.
4. Click the **All instances** tree item.
5. At the bottom of the Display Instances & Clusters screen, click **Add Instance**. An Add New Instance dialog box appears.
6. Select .NET from the list.
7. Select the application from the list.
If the desired application is not on the list, follow steps a-d:
 - a. Click **Add**.
 - b. In the Application Name field, type in the new application name.
 - c. On the Application System field, select the system application from the list.
 - d. Click **OK**.



To install an instance in an application, you must have ADMINISTRATE . INSTALL permission for the application in which you want to install the instance. If you only have permission for a specific technology, you will only be able to install in an application that is either empty, or is comprised of only the same technology.

8. Select a Tier from the list.
9. Click **OK** to view the next screen.



Always refresh AdminPoint after adding an instance.

Defining .NET property settings

To define properties in the “.NET Instance Installation-Properties Settings” screen

1. Click the drop-down menu and select the server with the .NET instance that you want to monitor. To add a new server, click **Add**. For more information, see see [Adding a new server](#).
 - ASP.NET instance
Select this option if you want to monitor a Web application. This instance type is only available if an Internet Information Server (IIS) is installed on the server.
 - a. Click the drop-down menu and select the instance you want to monitor.
 - b. At the bottom of the screen, enter a unique instance name to help you differentiate between instances.

- .NET instance
Select this option if you want to monitor other .NET processes.
 - a. Click the **browse (...)** button to select the instance you want to monitor.
 - b. At the bottom of the screen, enter a unique instance name to help you differentiate between instances.
 - c. Click **Next**.

About the AdminPoint Tier installation preview screen

The preview screen signifies that the AdminPoint Tier installer has gathered enough information to begin the installation phase.

The preview screen displays the following details before installation begins:

- Server name or IP address
- The name of the monitored server
- What actions will be performed in their respective order
- Installation folder (location)
- Required disk space
- Currently available disk space

At this screen, you can still change any of the settings displayed. Click **Back** to change any of the displayed settings. Click **Next** to continue installation. Click **Cancel** to stop the installation process.

About the AdminPoint Tier installation progress screen

This screen appears when the AdminPoint Tier installer is actually performing installation. Installation may take several minutes.

The progress screen visually displays the progress of actual installation on selected servers.

About the AdminPoint Tier installation complete screen

This screen appears when the AdminPoint Tier Installer has completed installation of all Tier instances.

If there are no action items, click **Finish** (it will be the only option available) to complete the instance installation. If there are action items, click **Next** (it will be the only option available).

About the AdminPoint Tier Action Items screen

After a Tier instance is installed, you must perform post-installation tasks as may be required. Only after completing any post-installation tasks required can you start using the Tier instances you have just installed.

Perform any tasks displayed on this screen. Click **Execute** for Precise to perform the action items. Click **Mark as done** if you have performed the action items yourself.

Post-installation action items

When the Microsoft .NET Tier Collectors have been installed, you need to perform post-installation action items. These action items vary depending on the .NET application type, and include editing configuration files and restarting Precise agents. Not all action items appear on screen as described in [Performing post-installation action items](#). This section includes instructions regarding action items that appear as part of the installation, as well as an overview of instrumentation action items that are not included.

Performing post-installation action items

To perform Microsoft .NET Tier post-installation tasks

1. In the Microsoft .NET Tier - Installation dialog box, click **Next**.
2. In the Microsoft .NET Tier - Post-Installation dialog box, follow the instructions to perform all post-installation tasks.
3. Click **Finish**.

Configuring DLLs, classes, and methods to instrument

You can add the DLLs you wish to monitor using the DLLs dialog box in the Instrumentation Settings menu, as explained in the Precise for Microsoft .NET User's Guide. But afterwards you must edit the file `instrumentation.xml` manually to define which classes and methods to instrument.

The Precise for Microsoft .NET activity tracking is based on instrumentation of your application DLLs. This information is configured in the `instrumentation.xml` file. The way you configure this file (this means which DLLs to instrument and which classes or methods) affect how much information will be tracked and also the overhead of the Precise for Microsoft .NET activity tracking on your application.

The `instrumentation.xml` file contains many configurable settings, which are explained in the [Precise Administration Guide](#) in detail.

The following file is an example section of the `instrumentation.xml` file which configures the instrumentation of the Microsoft PetShop sample Web application:

```

<instance name="AspNetIIS6">
    <dlls>
        <dll name="pet*.dll" /><!-- instrument all DLLs that start with pet* --!>
    </dlls>
    <instrument>
        <classes>
            <class>
                <name>Petshop.*</name><!-- instrument all classes of which the namespace begins with Petshop --!>
                <called-method>
                    <methods>
                        <method>
                            <name>*</name><!-- instrument all methods in each class --!>
                        </method>
                    </methods>
                </called-method>
            </class>
        </classes>
    </instrument>
...

```

Defining permission settings for a Precise installation folder

The Precise for Microsoft .NET hooks that are injected into the monitored .NET process, are run under the same user account on which your .NET application is running. This user must have full permission rights to the `<precise_root>` folder and all its sub-folders.

You can change the security settings of the `<precise_root>` folder with the Windows File Explorer Security tab.

 If you have several application pools which translate into different Windows users, all these users must be granted full permission to the `<precise_root>` folder.

Changing the activity tracking threshold

The Precise for Microsoft .NET installation pre-configures the activity data tracking threshold and sets it to 50 milliseconds. This means that only methods, service requests, SQL statements, and Web services that have a response time that is equal to, or greater than 50 milliseconds will be collected.

You can change this default value in the `ActivityCollector.xml` file, located in the `<precise_root>/products/dotnet/config` folder any time after the installation:

```

<activity-collector-config>
    <!-- psdn_act_col.exe configuration, configurable per server (default) & per instance -->
    <aggregator>
        <topnsql>5</topnsql><!-- Top n SQL statement to monitor -->
        <sla>5000</sla><!-- Red SLA breach value for top HTTP invocations in msec.-->
        <nearsla>1000</nearsla><!-- Yellow SLA breach value for top http invocations in msec. -->
        <insane-rt>300</insane-rt><!-- Long running thread timeout in seconds -->
        <bit-vector>false</bit-vector>
        <last-level-bit-vector>false</last-level-bit-vector>
    </aggregator>
    <!-- Tracker.dll settings, configurable per server (default) & per instance -->
    <tracker>
        <threshold>50</threshold><!-- Threshold in msec for filtering out events before forwarding it to
        the Collector agent. -->
    </tracker>
</activity-collector-config>

```

 The new setting will only take effect after your .NET application is restarted. For ASP.NET that means an IIS reset or recycling the application pool.

 This setting has a significant effect on the overall overhead of Precise for Microsoft .NET agent on your system. Lowering this value too much can cause your Web application and server to slow down considerably. If Precise for Microsoft .NET is installed on a development server, you can set this value to 0 to collect all data. On heavy loaded production systems, it is advisable not to change the default.

More information about controlling the Precise for Microsoft .NET overhead on your .NET application in the *Precise Administration Guide*.

About Precise for Microsoft .NET agent ports

The Precise for Microsoft .NET data collection is based on establishing a TCP connection between a tracking module (`tracker.dll`), the activity Collector and instrumentation agent. The default port numbers for the installation are respectively 20755 and 20756. If any of these port numbers are in use by another application on the server, you can change its value respectively in the `collector.xml` and `instrumenter.xml` files.

Port conflicts can cause the Precise for Microsoft .NET Collector not to work and to submit errors. The following is an error example that can appear in the dotnet.collector.activity.trc file:

```
=====
ID      : CONFIG-0000000
Time    : 2020-11-23 10:03:54.750 GMT+02 i3 Time    : 2020-11-23 08:03:54.750 GMT Process   : psdn_dncol_act.exe
-1 (ID:2704)
Logger   : com.Symantec.apm.i4dotnet.collector.activity
Msg      : Exception in remoting : System.Net.Sockets.SocketException: Only one usage of each socket address
(protocol/network address/port) is normally permitted
=====
```

The following is an error example that can appear in the dotnet.collector.dncol_isntr.trc file:

```
=====
ID      : CONFIG-0000000
Time    : 2020-11-23 10:09:05.842 GMT+02 i3 Time    : 2020-11-23 08:09:05.842 GMT Process   : psdn_dncol_instr.
exe (ID:4972)
Logger   : com.Symantec.apm.i4dotnet.collector.dncol_instr
Msg      : Unable to setup TCP server on port 20756, check that it is not used by another process, ex=Only one
usage of each socket address (protocol/network address/port) is normally permitted
=====
```

[IDERA](#) | [Products](#) | [Purchase](#) | [Support](#) | [Community](#) | [Resources](#) | [About Us](#) | [Legal](#)