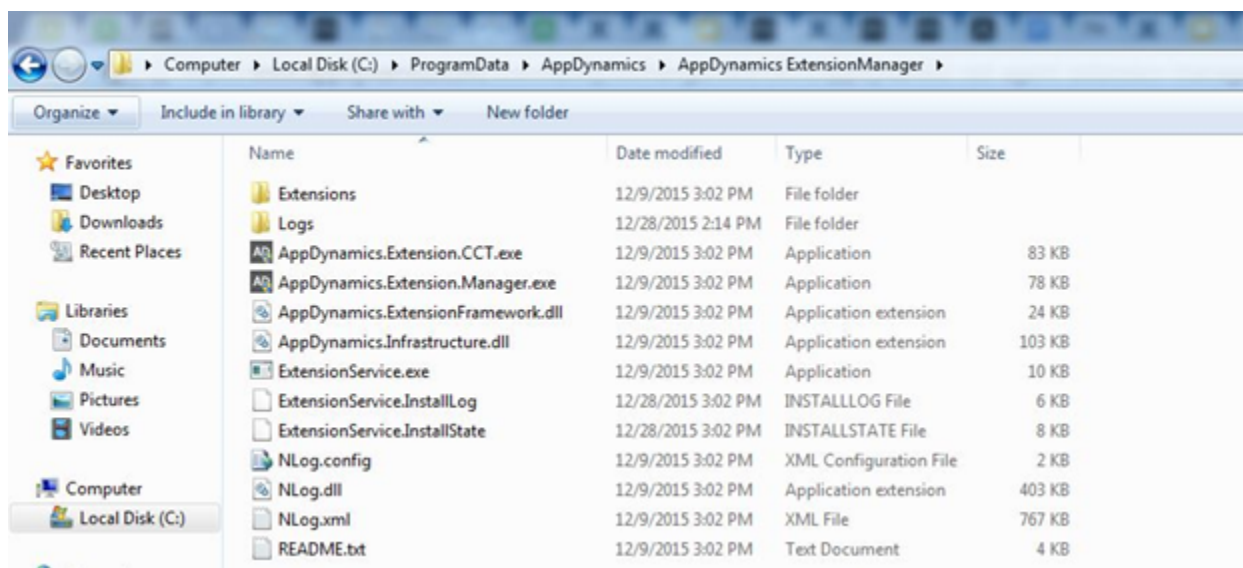
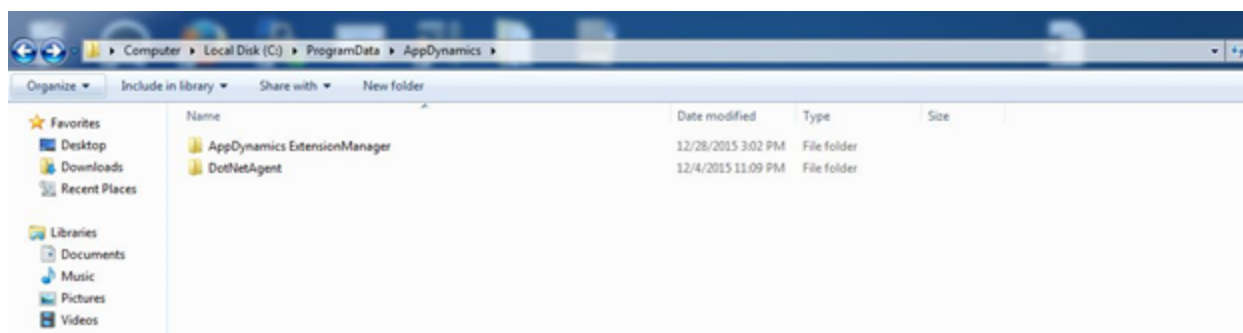


Windows Service Status Monitoring using Appdynamics .Net Extension Manager

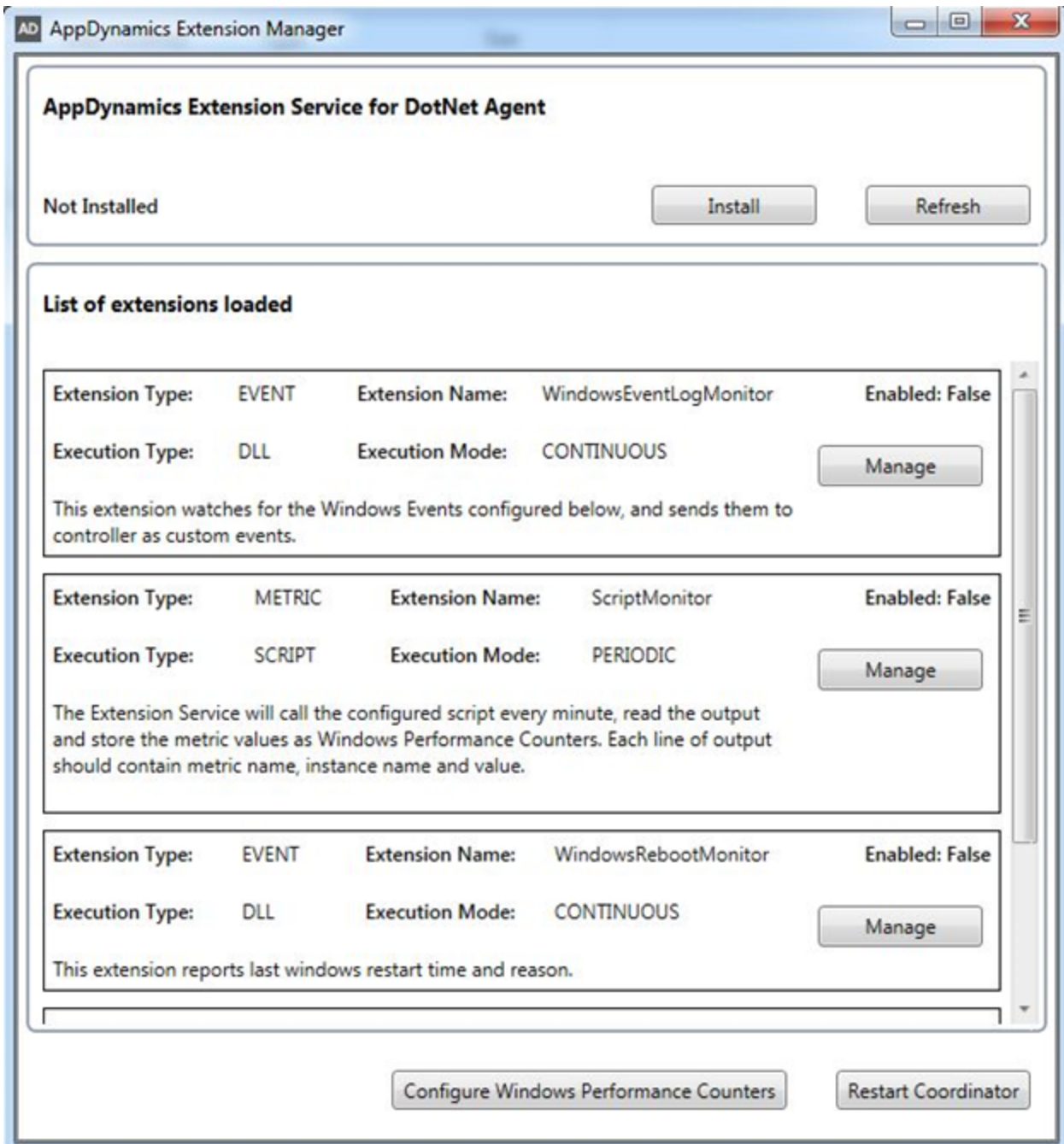
1. Download Extension Manager -

<https://www.appdynamics.com/community/exchange/extension/appdynamics-net-agent-extension-manager/>

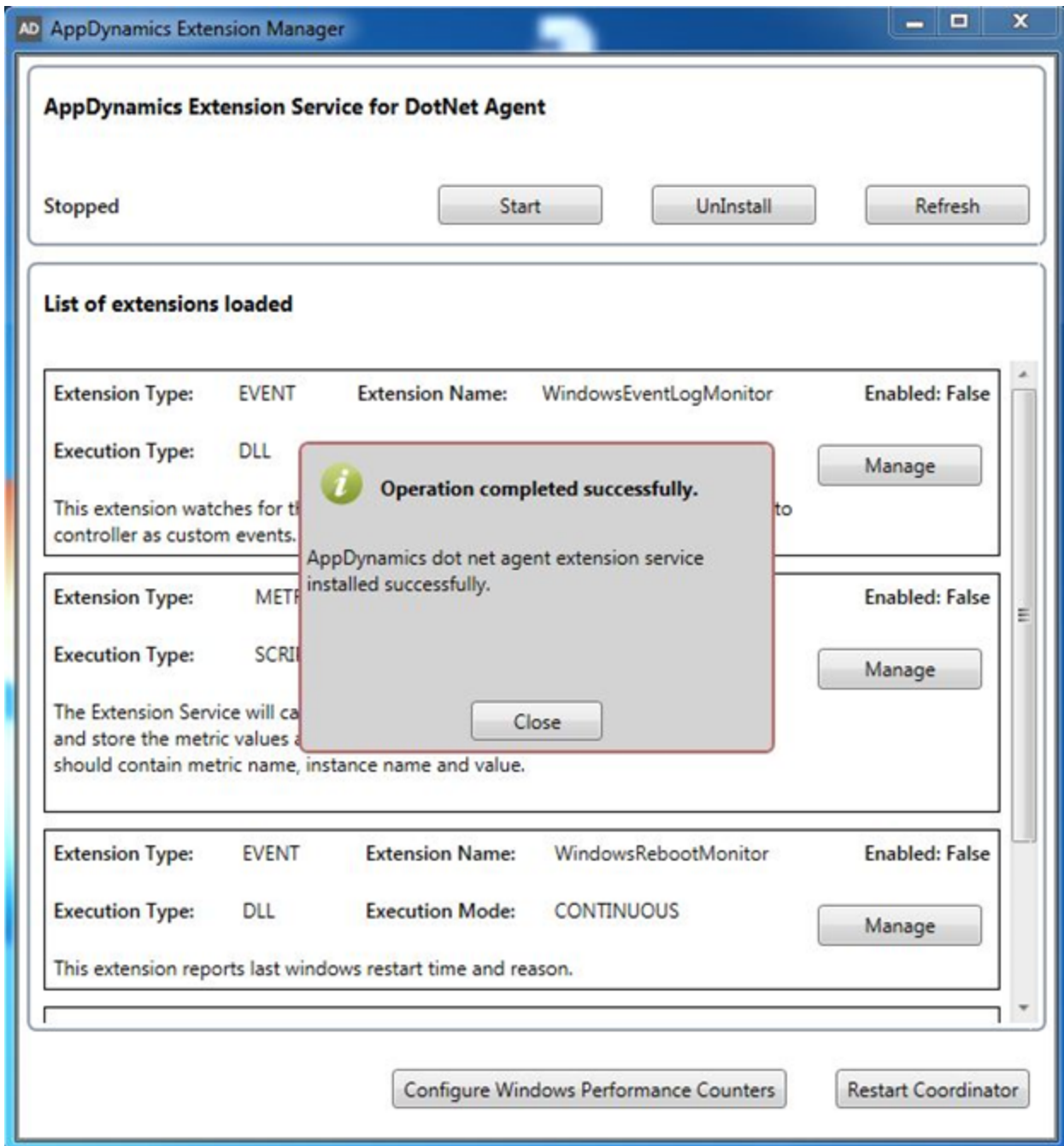
2. Unzip appdynamics-extensionmanager.zip under C:\ProgramData\AppDynamics



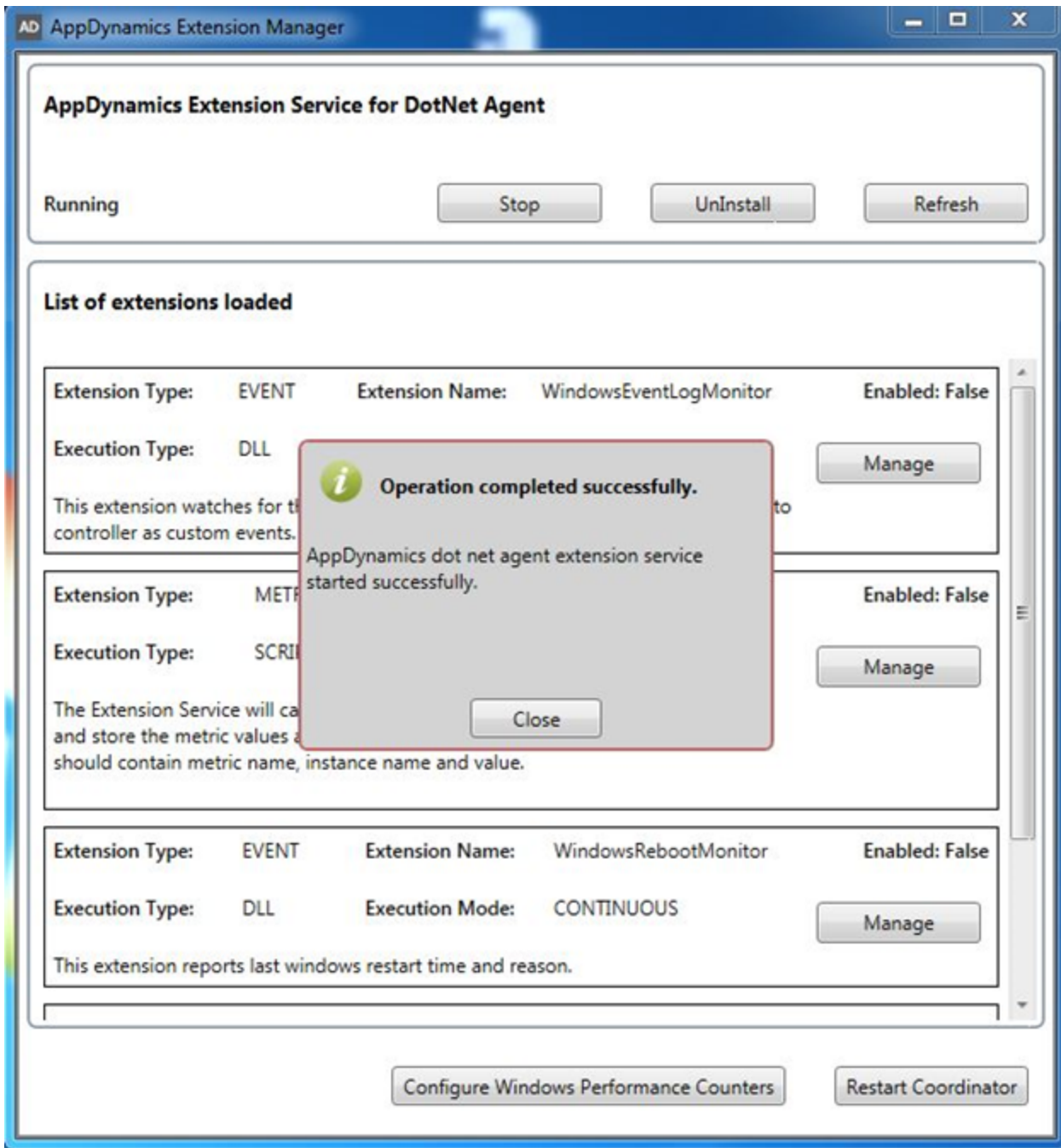
3. Run "AppDynamics.Extension.Manager.exe" as Administrator



4. Click on Install



5. Click on Start



6. Edit extension.xml file - C:\ProgramData\AppDynamics\AppDynamics ExtensionManager\Extensions\WindowsServiceStatusMonitor\extension.xml

```

<?xml version="1.0" encoding="utf-8" ?>
<!--
Metric Extensions create custom Windows Performance Counters to store metric data until it's sent to the Controller via the AppDynamics.Agent.Coordinator service.
The "name" property specified below is used to name this Extension, as well as the Performance Counter category these metrics will be added to.
-->
<extension type="Metric" name="WindowsServiceStatusMonitor" enabled="false">
  <execution mode="Periodic" type="DLL" frequency-in-seconds="60" path="WindowsServiceStatusMonitor" />
  <description>This extension can be used to monitor the status of any Windows service on the machine.</description>
<!--
As noted above, all "Metric" Extension metrics are stored as custom Windows Performance Counters on the server.
Performance Counters will be named after the metric name and instance names below, and stored in the "WindowsServiceStatusMonitor" Performance Counter Category as
specified by the Extension name.

Be sure to add these Performance Counters to the AppDynamics.Agent.Coordinator configuration file (config.xml) so that they're reported to the controller.
https://docs.appdynamics.com/display/PP04/Enable+Monitoring+for+Windows+Performance+Counters.
You can also use AppDynamics.Extension.CCT.exe included in your Extensions install folder to add these Performance Counters to your config.xml

The following metric, serviceStatus, will report the Windows service status as an integer value.
-->
<metrics>
  <metric name="ServiceStatus" enabled="true" description="Status of windows service" />
</metrics>
<!--
Instance name represents the name of the Windows service to be monitored.
-->
<instances>
  <instance name="AppDynamics.Agent.Coordinator_service" enabled="true" />
  <instance name="WCacheSvc" enabled="true" />
</instances>
</extension>

```

7. Change following sections

Enabled = true in the extension xml tag.

```

<?xml version="1.0" encoding="utf-8" ?>
<!--
Metric Extensions create custom Windows Performance Counters to store metric data until it's sent to the Controller via the AppDynamics.Agent.Coordinator service.
The "name" property specified below is used to name this Extension, as well as the Performance Counter category these metrics will be added to.
-->
<extension type="Metric" name="WindowsServiceStatusMonitor" enabled="true">
  <execution mode="Periodic" type="DLL" frequency-in-seconds="60" path="WindowsServiceStatusMonitor" />
  <description>This extension can be used to monitor the status of any Windows service on the machine.</description>
<!--
As noted above, all "Metric" Extension metrics are stored as custom Windows Performance Counters on the server.
Performance Counters will be named after the metric name and instance names below, and stored in the "WindowsServiceStatusMonitor" Performance Counter Category as
specified by the Extension name.

Be sure to add these Performance Counters to the AppDynamics.Agent.Coordinator configuration file (config.xml) so that they're reported to the controller.
https://docs.appdynamics.com/display/PP04/Enable+Monitoring+for+Windows+Performance+Counters.
You can also use AppDynamics.Extension.CCT.exe included in your Extensions install folder to add these Performance Counters to your config.xml

The following metric, serviceStatus, will report the Windows service status as an integer value.
-->
<metrics>
  <metric name="ServiceStatus" enabled="true" description="Status of windows service" />
</metrics>
<!--
Instance name represents the name of the Windows service to be monitored.
-->
<instances>
  <instance name="AppDynamics.Agent.Coordinator_service" enabled="true" />
  <instance name="WCacheSvc" enabled="true" />
</instances>
</extension>

```

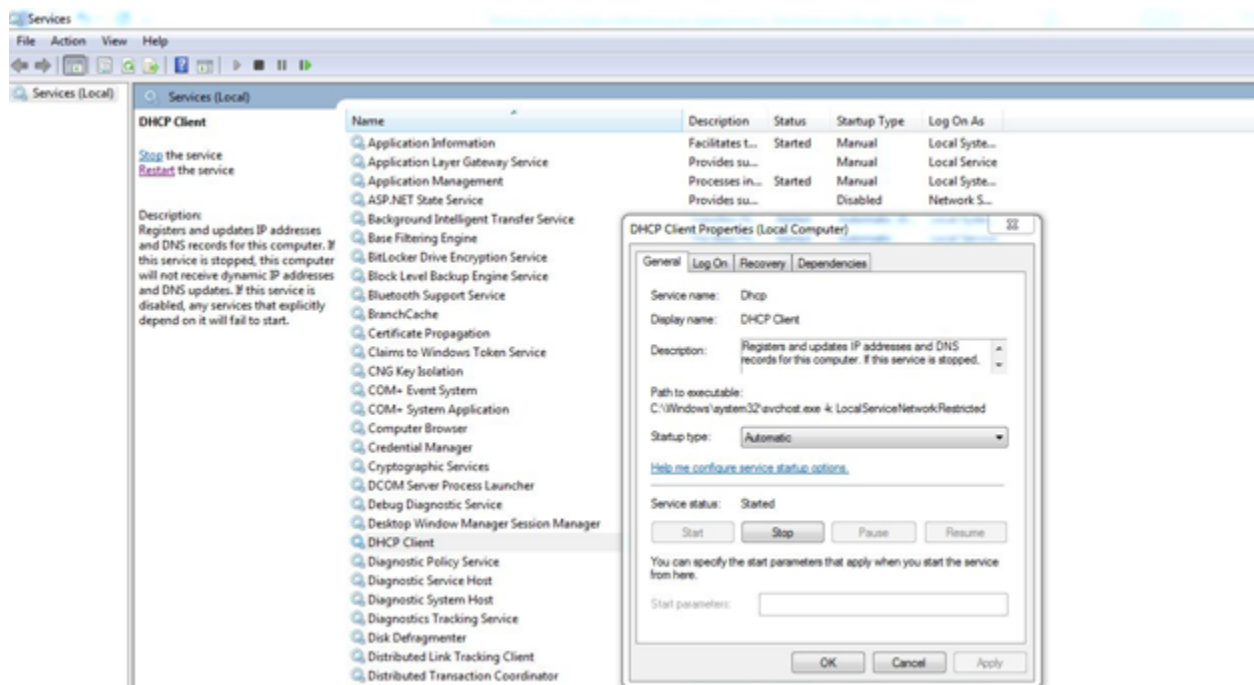
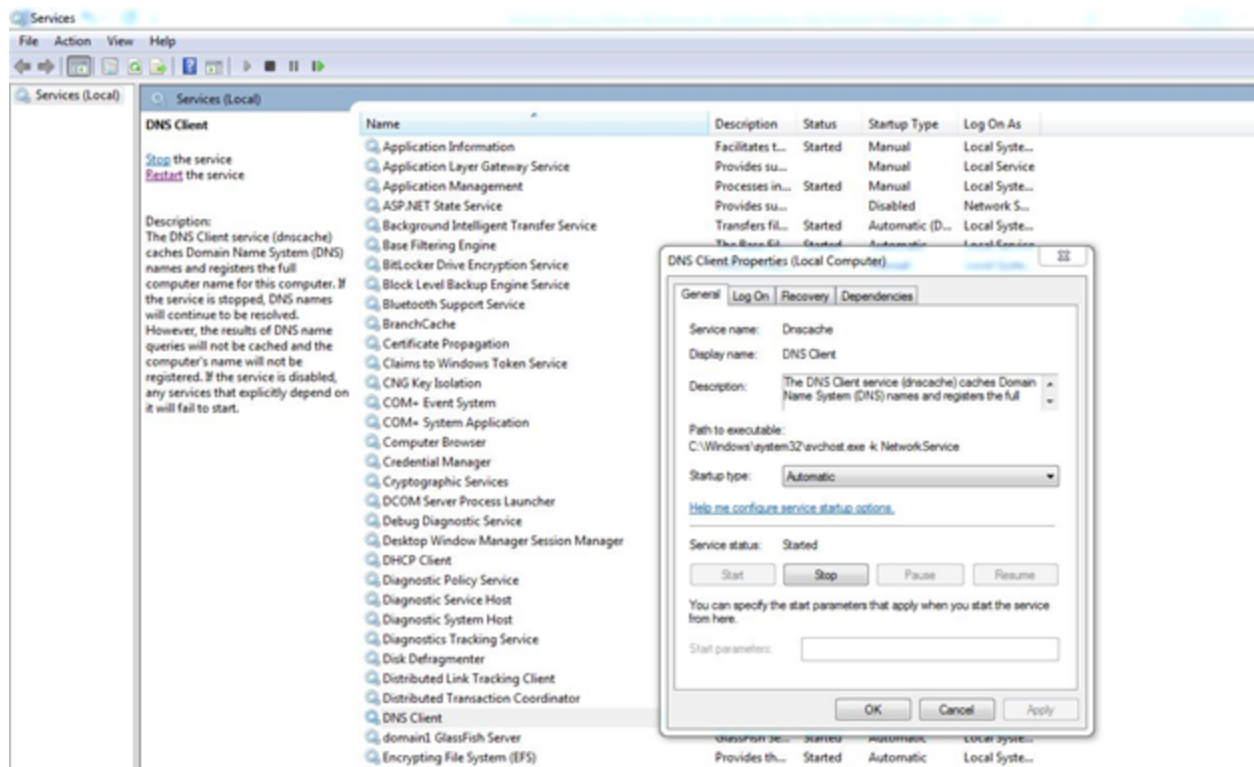
8. Add instance tag for all the windows services you want to monitor –

e.g.: You want to monitor DHCP Client service and DNS Client services.

a.) Get the service name from services.msc

DHCP Client Service Name: Dhcp

DNS Client Service Name : Dnscache



b.) Add following two entries under instances and save the file.

```
<instance name=" Dhcp" enabled="true" />
```

```
<instance name=" Dnscache" enabled="true" />
```

```

<?xml version="1.0" encoding="utf-8" ?>
<!--
Metric Extensions create custom Windows Performance Counters to store metric data until it's sent to the Controller via the AppDynamics.Agent.Coordinator service.
The "name" property specified below is used to name this Extension, as well as the Performance Counter category these metrics will be added to.
-->
<extension type="Metric" name="WindowsServiceStatusMonitor" enabled="true">
  <instance name="Periodic" type="Dtl" frequency-in-seconds="60" path="WindowsServiceStatusMonitor" />
  <description>This extension can be used to monitor the status of any Windows service on the machine.</description>
<!--
As noted above, all "Metric" Extension metrics are stored as custom Windows Performance Counters on the server.
Performance Counters will be named after the metric name and instance names below, and stored in the "WindowsServiceStatusMonitor" Performance Counter Category as
specified by the Extension name.

Be sure to add these Performance Counters to the AppDynamics.Agent.Coordinator configuration file (config.xml) so that they're reported to the controller.
https://docs.appdynamics.com/display/PS04/Enable+Monitoring+for+Windows+Performance+Counters.
You can also use AppDynamics.Extension.OCT.exe included in your Extensions install folder to add these Performance Counters to your config.xml

The following metric, serviceStatus, will report the Windows service status as an integer value.
-->
<metrics>
  <metric name="ServiceStatus" enabled="true" description="Status of windows service" />
</metrics>
<!--
Instance name represents the name of the Windows service to be monitored.
-->
<instances>
  <instance name="AppDynamics.Agent.Coordinator_service" enabled="true" />
  <instance name="DnscacheSvc" enabled="true" />
  <instance name=" Dhcp" enabled="true" />
  <instance name=" Dnscache" enabled="true" />
</instances>
</extension>

```

9. Restart following two services from services.msc

- AppDynamics.Agent.Extension_Service
- AppDynamics.Agent.Coordinator_service

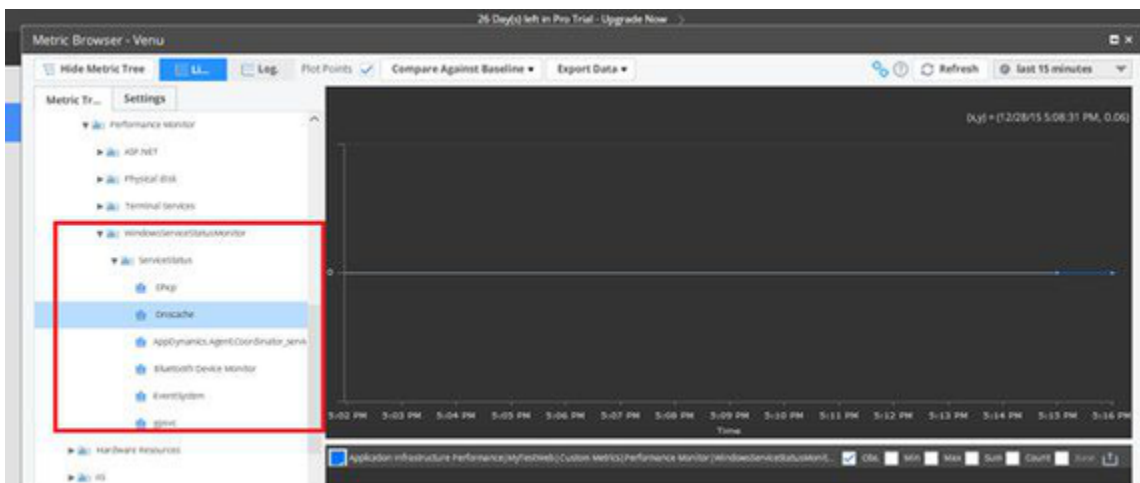
10. In the config.xml file located at C:\ProgramData\AppDynamics\Config, the **perfmom** counters should be added as mentioned below.

```

<machine-agent>
  <perf-counters>
    <perf-counter cat="WindowsServiceStatusMonitor" name="ServiceStatus" instance="AppDynamics.Agent.Coordinator_service" />
    <perf-counter cat="WindowsServiceStatusMonitor" name="ServiceStatus" instance="EventSystem" />
    <perf-counter cat="WindowsServiceStatusMonitor" name="ServiceStatus" instance="Bluetooth Device Monitor" />
    <perf-counter cat="WindowsServiceStatusMonitor" name="ServiceStatus" instance="gpsvc" />
    <perf-counter cat="WindowsServiceStatusMonitor" name="ServiceStatus" instance=" Dhcp" />
    <perf-counter cat="WindowsServiceStatusMonitor" name="ServiceStatus" instance=" Dnscache" />
  </perf-counters>
</machine-agent>

```

11. After the above steps, the custom metrics would appear in the **Metric Browser** at the following location (Application Infrastructure Performance|<tier name>|Custom Metrics|Performance Monitor|WindowsServiceStatusMonitor)



Different Service status and details -

- 1 : SERVICE_STOPPED
- 2 : SERVICE_START_PENDING
- 3 : SERVICE_STOP_PENDING
- 4 : SERVICE_RUNNING
- 5 : CONTINUE_PENDING
- 6 : PAUSE_PENDING
- 7 : SERVICE_PAUSED

So 1 means stopped and 4 means running.

Refer dwCurrentState section on [https://msdn.microsoft.com/en-us/library/windows/desktop/ms685996\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/windows/desktop/ms685996(v=vs.85).aspx) for more details