Step 3 - Connect Agent to Dynatrace

The final step to monitor your environment is to configure your application processes and webservers to load the Dynatrace AppMon Agent so that they connect to the Dynatrace AppMon Server. Dynatrace AppMon supports a variety of web and application servers and different technologies in heterogeneous environments.

YouTube Video Guides

In these videos we will walk you through a complete setup for typical applications environments either based on a Java, Microsoft .NET Environment, PHP stack or SharePoint.

Step-by-Step Guides

These step-by-step guides will walk you through connecting your environment to Dynatrace AppMon. Follow these steps closely and you will be up and monitoring in minutes.

For **SharePoint** please get started by downloading and applying the SharePoint FastPack
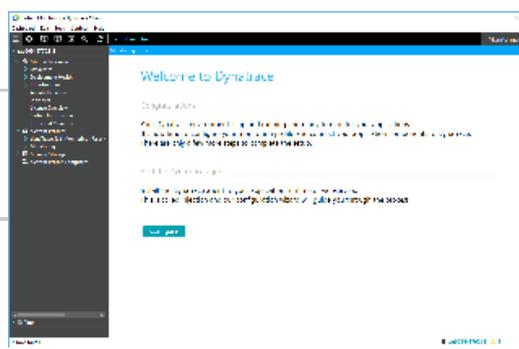
Webservers
Application Servers

| Apache HTTP |
| Apache with PHP |
| Microsoft IIS |
| NGINX |

The following steps describe how to configure and enable the Dynatrace AppMon Agent to monitor an Apache HTTPD webserver. This agent will not only allow you to retrieve webserver and host metrics but also enables you to use Dynatrace's User Experience Management functionality.

If this is the first agent you are adding after the Dynatrace setup is completed you should see the initial welcome

1. screen. Launch the configuration wizard by clicking on *Configure*. If you are already monitoring other agents then see the next step on how to launch the wizard again.
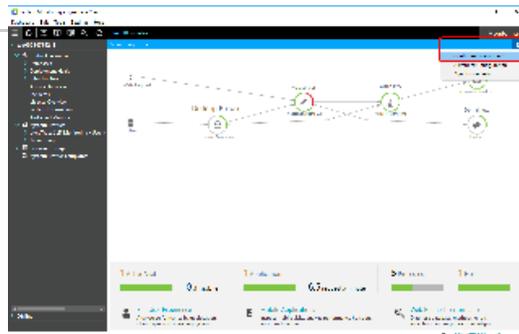
Launch the application environment configuration wizard through *Configure -> Configure Environment* and jump to step 4.

If you haven't done so yet, define the application type. In most cases you want to select the first option *Web / Java / .Net* here. This option generally fits well for typical multi-tier environments consisting

1. of frontend webservers and backend application servers.

To start configuring the agent for Apache HTTPD click on *Add* under the webserver symbol.

In the list of webserver types select Apache and provide a name for this tier (or just leave the default
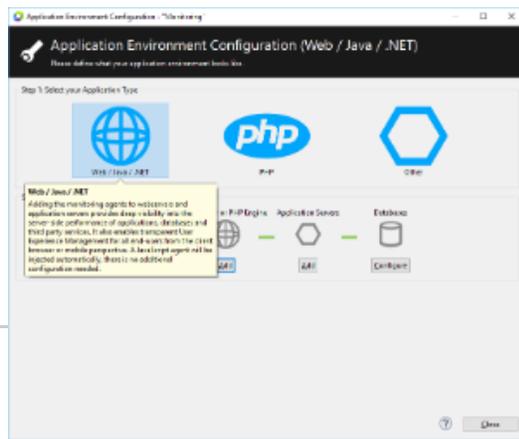
name). Confirm by clicking *Next*.

Now you need to specify where the webserver is running and which architecture it's using. This is important as we will need to install the right module for the Apache HTTPD in the next step. If you are uncertain you can check Apache's architecture by executing *httpd -V*.

```
# httpd -V
Server version: Apache/2.2.27 (Unix)
Server built:   Apr 25 2014 22:26:04
Server's Module Magic Number:
    20051115:33
    Server loaded:  APR 1.5.0,
    APR-Util 1.4.1
Compiled using: APR 1.5.0, APR-Util
1.4.1
Architecture:   64-bit
```

For MacOS: You can select Linux and get the download links for the actual agent from the links further down in the instructions

The wizard will then prompt you with the agent download fitting for your architecture. The agent will be downloaded from the web and you will need to log in with your credentials to access the download.

Alternatively you can download the agent installers here if you can't directly download it from within the client:

Webserver Agent for Apache Linux 64bit
Webserver Agent for Apache Linux 32bit
Webserver Agent for Apache Windows

For MacOS: We DO NOT Support Web Server on MacOS

Also the wizard will present how to install the agent on the target platform. We can copy the command line to install the webserver agent on our webserver machine.

After you have installed the Dynatrace AppMon Agent on your webserver machine we need to configure the agent to connect to the Dynatrace AppMon Server.

In the Dynatrace AppMon installation directory you will find a *dtwsagent.ini* file. Edit this file and change the two options as presented in your wizard. You can simply copy the settings from the wizard and post it or replace the default *dtwsagent.ini* file

The last step to connect Apache HTTPD with Dynatrace AppMon is to load the Dynatrace AppMon webserver module in Apache. Anywhere in your httpd.conf (or any other configuration loaded by Apache) you will need to place a *LoadModule* state ment to include the Dynatrace AppMon agent
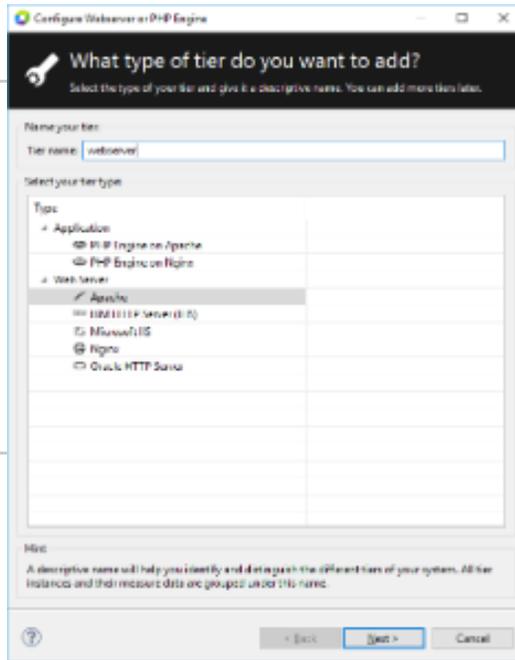
module during startup. Make sure you point to the right path of the *libdtagent.so* (libdtagent.dylib for Mac) in the *LoadModule* statement.
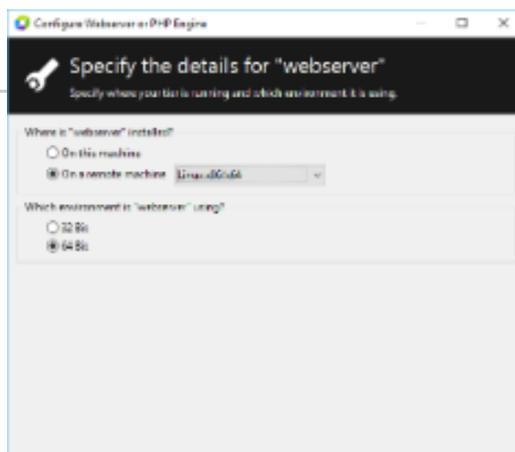
Now start or restart the *Dynatrace Webserver Agent* service. On Windows you can use the Services management console to do so. On Linux you can manage the service with the shipped init.d scripts:

```
# cd /opt/dynatrace-7.2
# ./init.d/dynaTraceWebServerAgent
start
```

After the webserver agent service is running restart the Apache webserver. Within a short time you should see the webserver agent connected to Dynat race AppMon and you can move on.

Back in the main dialog you now should also see the agent sucessfully connected. Also note that Inject JavaScript is enabled by default. This option will allow you to track end user experience (UEM) for webapplications from the browser side. For the full experience we recommend to keep this on.

1   You can now continue to add other agents, or close the wizard and start monitoring (don't forget to create some traffic on you webapplication).

When you are done with your Wizard you end up in the Welcome Screen. Now it is time to generate some load. Either access your application manually through e.g: Browser or run a short load testing script to generate some load. The Welcome Screen will immediately tell you how many transactions have already been captured. Once you have some traffic click on *Monitoring*.

The following steps describe how to configure and enable the Dynatrace AppMon agent to monitor an Apache webserver with PHP Engine. To enable monitoring of PHP powered applications we will first add the agent to Apache and then also enable it for
    the dynamic PHP execution. This will not only allow
    you to retrieve webserver and host metrics
    but also provide insight into PHP execution
and compile times. Additionally you will be able to use Dynatrace's user experience management functionality.

If this is the first agent you are adding after the Dynatrace setup is completed you should see the initial welcome screen. Launch the configuration wizard by clicking on *Configure*. If you are already monitoring other agents then see the next step on how to launch the wizard again.

Launch the application environment configuration wizard through *Configure -> Configure Environment*

and jump to step 4.

For most cases you should select the PHP Application type in the environment configuration wizard. If your environment is more complex than just an Apache Webserver with PHP enabled (e.g. additional Java Backends) you might want to select the general Web/Java/.NET application type.

In the next steps you will add an agent to the Apache webserver as well as an dedicated agent for the PHP engine. Click *Add* under the *PHP Engine on Webserver* icon.

Select *PHP Engine on Apache* and assign a name for this application tier or just leave the default.

Now you need to specify where the webserver is running and which architecture it's using. This is important as we will need to install the right module for the Apache HTTPD in the next step. If you are uncertain you can check Apache's architecture by executing *httpd -V.*

```
    # httpd -V
Server version: Apache/2.2.27 (Unix)
Server built:   Apr 25 2014 22:26:04
Server's Module Magic Number:
20051115:33
Server loaded:  APR 1.5.0, APR-Util
1.4.1
Compiled using: APR 1.5.0, APR-Util
1.4.1
Architecture:   64-bit
```

For MacOS: You can select Linux and get the download links for the actual agent from the links further down in the instructions

The wizard will then prompt you with the agent download fitting for your architecture. The agent will be downloaded from the web and you will need to log in with your credentials to access the download.

Alternatively you can download the agent installers here if you can't directly download it from within the client:

Webserver Agent for Apache Linux 64bit
Webserver Agent for Apache Linux 32bit
Webserver Agent for Apache Windows

Webserver Agent for MacOS 32Bit
Webserver Agent for MacOS 64Bit

Also the wizard will present how to install the agent on the target platform. We can copy the command line to install the webserver agent on our webserver machine.

After you have installed the Dynatrace AppMon agent on your webserver machine we need to configure the agent to connect to the Dynatrace AppMon server.

In the Dynatrace AppMon installation directory (*<dyn atrace install path>/agent/conf*) you will find a *dtwsa*

gent.ini file. Edit this file and change the two options

as presented in your wizard. You can simply copy the settings from the wizard and post it or replace the default *dtwsagent.ini* file

Now we need to first enable the Dynatrace AppMon Agent for Apache HTTP and then also for PHP. To connect Apache HTTPD with Dynatrace AppMon we need to load the Dynatrace AppMon webserver module in Apache. Anywhere in your httpd.conf (or any other configuration loaded by Apache) you need to place a *LoadModule* statement to include the Dynatrace AppMon Agent module during startup. Make sure you point to the right path of the *libdtage nt.so* (libdtagent.dylib for mac) in the *LoadModule* st atement.

```
LoadModule dtagent_module
    /opt/dynatrace-7.2/agent/lib64/lib
    dtagent.so
```

Then edit your php.ini file and add a new dynamic extension.

```
extension=/opt/dynatrace-7.2/agent/lib
64/libdtagent.so
```

Now start or restart the D*ynatrace Webserver Agent* service. On Windows you can use the Services management console to do so. On Linux you can manage the service with the shipped init.d scripts:

```
# cd /opt/dynatrace-7.2
# ./init.d/dynaTraceWebServerAgent
start
```


After the webserver agent service is running restart the Apache webserver. Within a short time you should see the webserver agent connected to Dynatrace AppMon and you can move on.

After both services have been restarted you should see 2 Agents connected.

Finish the installation and access your PHP application to create some traffic.

Back in the main dialog you now should also see the agents sucessfully connected. Also note that Inject JavaScript is enabled by default. This option will allow you to track end user experience (UEM) for webapplications from the browser side. For the full experience we recommend to keep this on.

You are now ready to monitor your PHP application and should create some traffic, click through the main usecases and Dynatrace will start monitoring.

When you are done with your Wizard you end up in the Welcome Screen. Now it is time to generate some load. Either access your application manually through e.g: Browser or run a short load testing script to generate some load. The Welcome Screen will immediately tell you how many transactions have already been captured. Once you have some traffic click on *Monitoring*.
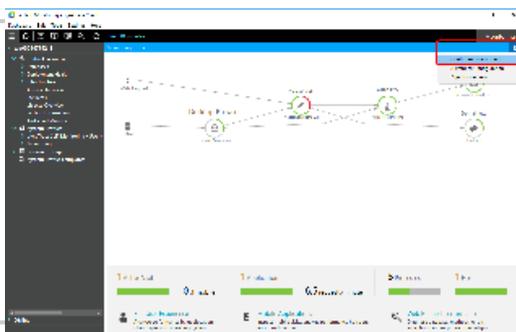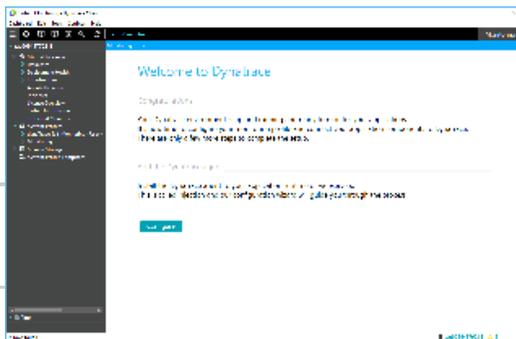
The following steps describe how to configure and enable the Dynatrace AppMon webserver agent for Microsoft's Internet Information Services. At this point it's important to know that if you are also using IIS to serve dynamic content created via ASP.NET you will also have to add the Dynatrace AppMon Agent to the .NET runtime (the application server part) of IIS. This part of the configuration will only cover the webserver part.

If this is the first agent you are adding after the Dynatrace setup is completed you should see the initial welcome screen. Launch the configuration wizard by clicking on *Configure*. If you are already monitoring other agents then see the next step on how to launch the wizard again.

Launch the application environment configuration wizard through *Configure -> Configure Environment* and jump to step 4.

If you haven't done so yet, define the application type. In most cases you want to select the first option *Web / Java / .Net* here. This option generally fits well for typical multi-tier environments consisting of frontend webservers and backend application servers.

To start configuring the agent for Microsoft's IIS click on *Add* under the webserver symbol.

In the list of webserver types select Microsoft IIS and provide a name for this tier (or just leave the default name). Confirm by clicking *Next*.

Now you need to specify where the webserver is running and which architecture it's using. In the next step we will download the right installer for this platform.

The wizard will then prompt you with the agent installer download fitting for your architecture. The agent will be downloaded from the web and you will need to log in with your credentials to access the download.

Alternatively you can download the agent installers here if you can't directly download it from within the client:

Webserver Agent for IIS Windows

Once the installer has been downloaded copy it to the machine (if not the local one) where Microsoft IIS is running.

After you have installed the Dynatrace AppMon Agent on your webserver machine we need to configure the agent to connect to the Dynatrace AppMon server.

In the Dynatrace AppMon installation directory (*C:\Program Files\dynatrace\dynatrace x.y\agent\conf*) you will find a *dtwsagent.ini* file. Edit this file and change the two options as presented in your wizard. You can simply copy the settings from the wizard and paste it at the beginning or replace the default *dtwsagent.ini* file.

To activate the Webserver Agent in IIS we will have to launch the IIS Management Console (*inetmgr.exe*) to enable the agent module there.

In the IIS Manager Console navigate to your server (or if you want to enable the agent only for a specific site select only the site) and enter the *Module* configuration.

In the *Modules* configuration go to *Configure Native Modules* to add the Dynatrace AppMon webserver agent module.

From the list of registered modules select both Dynatrace AppMon modules (32bit and 64bit). Regardless of the platform you are using you can safely enable both only the right one will be loaded and enabled. Confirm the setting.

Now first restart the *dynaTrace Webserver Agent Service.* You can either do this from the Windows Services or in a command shell by executing (note that you need administrative privileges to do that):

```
C:\>sc stop "Dynatrace Web Server
Agent 7.2"
C:\>sc start "Dynatrace Web Server
Agent 7.2"
```

After the Dynatrace AppMon Web Server Agent service has been started return to the IIS Management Console and restart IIS or use the command shell to restart the W3SVC service

```
C:\>sc stop w3svc
    C:\>sc start w3svc
```

Finally browse to your site so that the agent connects to Dynatrace!

Back in the configuration wizard you should now see the agent for IIS connected. You can confirm the dialog and return to the main screen of the wizard.

Back in the main dialog you now should also see the webserver agent sucessfully connected. Also note that *Inject JavaScript* is enabled by default. This option will allow you to track end user experience (UEM) for webapplications from the browser side. For the full experience we recommend to keep this on.

You can now continue to add other agents, or close the wizard and start monitoring (don't forget to create some traffic on your web application).

When you are done with your Wizard you end up in the Welcome Screen. Now it is time to generate some load. Either access your application manually through e.g: Browser or run a short load testing script to generate some load. The Welcome Screen will immediately tell you how many transactions have already been captured. Once you have some traffic click on *Monitoring*.

The following steps describe how to configure and enable the Dynatrace AppMon agent to monitor an NGINX webserver. This agent will not only allow you to retrieve webserver and host metrics but also enables you to use Dynatrace's User Experience Management functionality.

If this is the first agent you are adding after the Dynatrace setup is completed you should see the initial welcome screen. Launch the configuration wizard by clicking on *Configure*. If you are already monitoring other agents then see the next step on how to launch the wizard again.

Launch the application environment configuration wizard through *Configure -> Configure Environment* and jump to step 4.

If you haven't done so yet, define the application type. In most cases you want to select the first option *Web / Java / .Net* here. This option generally fits well for typical multi-tier environments consisting of frontend webservers and backend application servers.

To start configuring the agent for NGINX click on *Ad d* under the webserver symbol.

In the list of webserver types select Nginx and provide a name for this tier (or just leave the default name). Confirm by clicking *Next*.

Now you need to specify where the webserver is running and which architecture it's using. This is important as we will need to install the right agent for this architecture.

The wizard will then prompt you with the agent download fitting for your architecture. The agent will be downloaded from the web and you will need to log in with your credentials to access the download.

Alternatively you can download the agent installers here if you can't directly download it from within the client:

[Webserver Agent for NGINX L](#)inux 64bit
Webserver Agent for NGINX Linux 32bit

Also the wizard will present how to install the agent on the target platform. We can copy the command line to install the webserver agent on our webserver machine.

After you have installed the Dynatrace AppMon agent on your webserver machine we need to configure the agent to connect to the Dynatrace AppMon server.

In the Dynatrace AppMon installation directory (<dynatyrace install directory>/agent/conf) you will find a *dtwsagent.ini* file. Edit this file and change the two options as presented in your wizard. You can simply copy the settings from the wizard and post it or replace the default *dtwsagent.ini* file

To connect NGINX with Dynatrace AppMon you need to start it with the LD_PRELOAD option. Make sure you point LD_PRELOAD to the location of libdtagent.so (<dynatrace install directory>/agent/lib64/libdtagent.so). Test by starting nginx with

```
#
LD_PRELOAD=<install>/agent/lib64/l
ibdtagent.so nginx
```

Later you might want to add the LD_PRELOAD to your nginx startup script.

Now start or restart the D*ynatrace Webserver Agent* service. On Linux you can manage the service with the shipped init.d scripts:

```
# cd <dynatrace install directory>
# ./init.d/dynaTraceWebServerAgent
start
```

In the configuration wizard you should now see the Dynatrace webserver agent and Nginx connected.

You can close the summary screen of the wizard and return to the main page.

You can continue to add more agent/tiers to your environment. If you are using Nginx as a loadbalancer or proxy you might want to add your application servers or other webserver types. Also note that Inject JavaScript is enabled by default. This option will allow you to track end user experience (UEM) for webapplications from the browser side. For the full experience we recommend to keep this on.

When you are done with your Wizard you end up in the Welcome Screen. Now it is time to generate some load. Either access your application manually through e.g: Browser or run a short load testing script to generate some load. The Welcome Screen will immediately tell you how many transactions have already been captured. Once you have some traffic click on *Monitoring*.

The following steps describe how to configure and enable the Dynatrace AppMon agent to monitor a Java based application server. In this example we will be using Apache Tomcat but the steps are mostly identical for other Appservers like JBoss or Websphere. With the agent enabled in your application server you will not only get deep insight into the application execution but also get basic host monitoring for the machine it's running on.

If this is the first agent you are adding after the Dynatrace setup is completed you should see the initial welcome screen. Launch the configuration wizard by clicking on *Configure.* If you are already monitoring other agents then see the next step on how to launch the wizard again.

Launch the application environment configuration wizard through *Configure -> Configure Environment* and jump to step 4.

If you haven't done so yet, define the application type. In most cases you want to select the first option *Web / Java / .Net* here. This option generally fits well for typical multi-tier environments consisting of frontend webservers and backend application servers.

To start configuring the agent for the application server (Apache Tomcat in this case) click on *Add* un der the application servers symbol.

Select your application server from the list and provide a name for this tier (or just leave the default name).
In this case we are using Apache Tomcat as our Java application server. As we will see in the next steps, the key for adding the agent is to provide an additional parameter to the Java Virtual machine running the application server.
This will be done by modifying the server's start script. Confirm your selection by clicking *Next.*

Now you need to specify on which OS the application server is running and which architecture it's using. If your application server is running on the same machine as the Dynatrace server (only recommended for test or demo environments) the agent is already installed and you can skip the next step.

Make sure you select the right Java Virtual Machine version, as a 32bit agent will not work with a 64bit JVM and vice versa. If you are unsure which JVM you are using verify by calling:

```
# java -version
java version "1.7.0_21"
Java(TM) SE Runtime Environment (build
1.7.0_21-b12)
Java HotSpot(TM) 64-Bit Server VM
    (build 23.21-b01)
```

If your application server is running on another machine than the Dynatrace AppMon server you will need to download the correct agent installers from the web and copy them to the machine and install them there.

Alternatively you can download the agent installers here if you can't directly download it from within the client:

[Java Agent for Linux 32bit and 64bit JVMs](#)
[Java Agent for Windows 32bit and 64bit JVMs](#)

The wizard will present how to install the agent on the target platform. For windows based installations execute the installer package and follow the instructions of the installer. On Linux platforms you will need a java virtual machine to execute the JAR installer.

Java Agent for MacOS!!

For MacPS the Wizard doesn't give you any download instructions. Please just select Linux to keep going with the wizard.

The download for MacOS Agents for your JVMs are here: [32bit JVMs](#) and [64bit JVMs](#)

Once you have installed the Dynatrace AppMon agent on your server you need to add the *-agentpath* parameter to the Java Virtual Machine startup commandline. The wizard has already created the right parameter string for you. You only need to replace the *<path>* placeholder with the actual installation path of the Dynatrace agent. Again, make sure you are using the correct path to the *libdtagent.so* library (libdtagent.dylib for MacOs). Especially pay attention to the 32/64bit library directories, so that they match your JVM architecture.

For Apache Tomcat you will need to edit the catalina.sh file and modify the *JAVA_OPTS* variable to include the *-agentpath* parameter. If you are unsure how to modify your application server's startup configuration you might find more guidance in our [documentation](#).

For MacOS:

Here is an example agent path for the MacOS users. The wizard is not yet giving you the correct path - but I guess you will figure it out once you see this sample path: -agentpath:/YOURDIR/dynatrace-6.5/agent/lib64/libdtagent.dylib=name=YOURTIER_Monitoring,server=YOURDYNATRACESERVER:9998

Restart your application server and it should connect to the Dynatrace AppMon server. This could take up to a minute. If your application servers do no show up check the application server's log for any warnings or errors.

If you want to connect multiple application servers with the same configuration (e.g. the single nodes of

a cluster), configure all servers as described in step 8 and restart them. Once you see all servers connected, proceed and finish the wizard.

Back in the main dialog you should now see the application server agents sucessfully connected. Unless you see any warnings there you are ready to start monitoring or optionally add more tiers (like additional application servers).

When you are done with your Wizard you end up in the Welcome Screen. Now it is time to generate some load. Either access your application manually through e.g: Browser or run a short load testing script to generate some load. The Welcome Screen will immediately tell you how many transactions have already been captured. Once you have some traffic click on *Monitoring*.

The following steps describe how to configure and enable the Dynatrace AppMon agent to monitor a .Net based application server. In this example we will be using Microsoft IIS but the steps are mostly identical for other .Net processes like MS Webmatrix development environment. With the agent enabled in your application server you will not only get deep insight into the application execution but also get basic host monitoring for the machine it's running on.

If this is the first agent you are adding after the Dynatrace setup is completed you should see the initial welcome screen. Launch the configuration wizard by clicking on *Configure*. If you are already monitoring other agents then see the next step on how to launch the wizard again.

Launch the application environment configuration wizard through *Configure -> Configure Environment* and jump to step 4.

If you haven't done so yet, define the application type. In most cases you want to select the first option *Web / Java / .Net* here. This option generally fits well for typical multi-tier environments consisting of frontend webservers and backend application servers.

To start configuring the agent for the application server (Microsoft IIS/.Net) click on *Add* under the application servers symbol.

Select Microsoft IIS/ASP.NET from the list and provide a name for this tier (or just leave the default name). In this case we want to monitor the DotNetNuke example application so we are using DotNetNuke as the tier name. If you have already configured the webserver agent for IIS do not get confused. IIS acts both as a webserver but also an application server. Since we

want to monitor the .NET part we have to add the

agent to the worker processes for .NET (w3wp.exe) in the next steps. Confirm your selection by clicking *Next*.

Specify where your application is running and which architecture you are using. This will ensure to provide the right downloads for the agent installer in the next step.

If your application server is running on another machine than the Dynatrace AppMon server you will need to download the correct agent installers from the web and copy them to the machine and install them there.

Alternatively you can download the agent installer here if you can't directly download it from within the client:

.Net Agent for Windows 32bit and 64bit JVMs

Execute the installer package and follow the instructions.

Once you have installed the Dynatrace AppMon agent on your server you need to add the agent to the .NET runtime for the application server process (w3wp.exe) The wizard has already created the right settings (Agent Name, server and port) for you.
To configure the .NET runtime you need to launch the Agent Configuration Tool which has been installed together with the Agent. If the .NET server is running on the same machine as the Dynatrace AppMon environment click on *Start Agent Configuration Tool* to launch it. Otherwise launch the *Agent Configuration Tool* on the machne your IIS is running on. The config tool will need administrative privileges to apply the changes.

Launch the Dynatrace AppMon Agent Configuration for .Net tool from the start menu (Start - > Dynatrace Agent Configuration for .Net). Click on the plus sign to create a new configuration.

From the list of processes pick the IIS worker process (*w3wp.exe*). If w3wp.exe is not listed here make sure you have started IIS and created som requests to your application. Otherwise the application might be idle and no worker processes might be active at this time.

Note: if you are not using IIS but any other .Net process (e.g. you might be using MS Webmatrix only for developing your .Net application) you can select that process as well.

Click next to define the Dynatrace AppMon settings for this process.

Now copy the Agent Name from the configuration wizard and also make sure to set the dynaTrac server/collector address and port.

Click on Finish to return to the agent configuration overview.

In the Agent Configuration Overview you should now see your just created Agent Configuration. Make sure the agent is active and then restart /recycle IIS. After accessing the application again you should see the runningworker process in the lower section of the dialog. The Status column should now read .Net agent successfully loaded.

Switch back to the Dynatrace AppMon Client configuration screen and you should see the connected agent showing up shortly.

After the agent has conencted to Dynatrace AppMon click Next to finish the installation.

The wizard will present you a summary of what you have just configured. Close the dialog to return to the main overview.

In the Application Environment Configuration you should now see the .Net process (DotNetNuke here) connected as well as the webserver agent of IIS if you have already configuraed that. Please note that the webserver agent is a seperate agent from the .Net agent. Although both are running in IIS they are technically two different things.

You can now continue to add more agents or close the configuration wizard and start using Dynatrace AppMon.

When you are done with your Wizard you end up in the Welcome Screen. Now it is time to generate some load. Either access your application manually through e.g: Browser or run a short load testing script to generate some load. The Welcome Screen will immediately tell you how many transactions have already been captured. Once you have some traffic click on *Monitoring*.

After you have connected all your web- and application servers you are ready to use and explore dynaTrace!

Next Step: **Use and Explore**

Java Based Application

Microsoft IIS/.Net