Oracle® Cloud

Installing and Configuring Oracle Application Performance Monitoring

E89051-27
December 2020

Oracle Cloud Installing and Configuring Oracle Application Performance Monitoring,

E89051-27

Primary Author: Oracle Corporation

Contents

11 Install and Configure APM Ruby Agent

12 Install and Configure APM Agents on Containers

13 Troubleshoot the Deployment of Application Performance Monitoring

14 Set Up End User Monitoring

15 Configure Data Collection and Privacy Controls

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

1

Introduction to Application Performance Monitoring

Oracle Application Performance Monitoring is a Platform as a Service (PaaS) based solution that provides deep visibility into the performance of your application, from end user to application logs. The service integrates user experience information and application metrics along with log data analytics.

Topics

- Overview
- Types of APM Agents

Overview

Today's e-businesses depend heavily on their web applications to allow critical business processes to be performed online. As more emphasis is placed on accessing information quickly, remotely, and accurately, you should take proactive steps to ensure that your online customers can successfully complete a transaction. Application Performance Monitoring is a cloud service that provides deep visibility into the performance of your web application.

With Application Performance Monitoring, you can:

- Rapidly isolate application performance issues.
- Drill down to related logs in context of a problem and find its root cause.
- Gain end-to-end visibility into the performance of your application across all tiers.
- Monitor end-user experience.

Take a tour on what you can do with the Application Performance Monitoring Product Tour.

Types of APM Agents

Oracle Management Cloud supports various APM Agents.

Follow these instructions to install an APM Agent for your environment.

APM Agent	Installation Instructions
APM Java Agent on WebLogic Server	Install APM Java Agent on WebLogic Server
APM Java Agent on WebSphere Server	Install APM Java Agent on WebSphere Server

APM Java Agent on Apache Tomcat Server	Install APM Java Agent on Apache Tomcat Server
APM Java Agent on JBoss Server	Install APM Java Agent on JBoss Server
APM Java Agent on Jetty Server	Install APM Java Agent on Jetty Server
APM Java Agent on Peoplesoft	Install APM Java Agent on Peoplesoft
APM Java Agent on Oracle E-Business Suite	Install APM Java Agent on Oracle E-Business Suite
APM .Net Agent	Install APM .Net Agent
APM Node.js Agent	Install APM Node.js Agent
APM Ruby Agent	Install APM Ruby Agent
APM Java Agent on Docker	Install APM Java Agent on Docker
APM Node.js Agent on Docker	Install APM Node.js Agent on Docker
APM Ruby Agent on Docker	Install APM Ruby Agent on Docker

Licensing Overview

Application Performance Monitoring is included with Oracle Management Cloud - Standard Edition as well as Oracle Management Cloud - Enterprise Edition. For more details on all Oracle Management Cloud offerings, see Oracle Management Cloud Offerings in *Getting Started with Oracle Management Cloud.*

Oracle Management Cloud subscribes to the Oracle Cloud Universal Credits model, a flexible buying and usage model for Oracle Cloud services. To use Application Performance Monitoring, you need to enable either Oracle Management Cloud – Standard Edition or Oracle Management Cloud – Enterprise Edition. For more details, see Oracle Management Cloud License Information in *Getting Started with Oracle Management Cloud*. To see how entities are calculated when you use Application Performance Monitoring, see Entities Conversion Formulas in *Getting Started with Oracle Management Cloud*.

For details of licensing of Oracle Management Cloud deployed on a Traditional Cloud Account, see Traditional Cloud Account in *Getting Started with Oracle Management Cloud.*

Oracle Management Cloud is also available in the Government Subscription Model, which is specifically designed for national and local governments. See Oracle Management Cloud – Government Subscription Model in *Getting Started with Oracle Management Cloud.*

2

Install and Configure APM Java Agent On Oracle Weblogic Server

Topics:

- On Linux: Requirements and Installation Instructions
- On Windows: Requirements and Installation Instructions

On Linux: Requirements and Installation Instructions

Prerequisites for Deploying APM Java Agent on an Oracle WebLogic Server

- Supported versions of Oracle WebLogic Server:
  - Oracle WebLogic Server 12.1.3
  - Oracle WebLogic Server 12.2.1
  - Oracle WebLogic Server 10.3.6, also known as Oracle WebLogic Server 11*g* Release 1

> **Note:**
>
> In the host running Oracle WebLogic Server, you can run the following command from `$WLS_HOME/server/lib` to check the Oracle WebLogic Server version:
>
> ```
> java -cp weblogic.jar weblogic.version
> ```

- JDK version 1.7 or above. If you are using JDK 1.6, either use a JDK 1.6 that supports TLS 1.2 security protocol, or connect to OMC through a Gateway.
- If the JDK version you are running doesn't support TLS 1.2 security protocol, refer to My Oracle Support Doc ID 2703411.1 before proceeding with the APM agent installation.

Other considerations:

- The machine hosting the Oracle WebLogic Server should be able to establish an HTTPS connection either directly or indirectly (using a proxy server or an Oracle Management Cloud gateway) to Oracle Management Cloud. For more information about Oracle Management Cloud gateway, see Install a Gateway.
- The HTTPS connection must use TLS 1.2 security protocol.
- The install user of APM Java Agent should be the same as the Oracle WebLogic Server user.

- The Oracle WebLogic Server user should have read and write permissions to the directories that host the APM Java Agent.

Licensing considerations:

- If you are installing Application Performance Monitoring on the WebLogic Administration Server, then it will also be automatically installed on all the managed servers. One APM Agent will run on each of your managed servers, and this should be considered while calculating the total licensing cost.
- If you prefer to run the APM Agent on one or a few selected managed servers only, then install Application Performance Monitoring on only those managed servers.

Set the `DOMAIN_HOME` Variable

Set the `DOMAIN_HOME` variable to point to the Oracle WebLogic Server domain.

Example:

```
export DOMAIN_HOME=<WebLogic Server Domain>
```

Deploy a Gateway (Optional)

Gateway is not a mandatory component while deploying Oracle Application Performance Monitoring; you can use a gateway in the following scenarios:

- If you have an application server that does not support Transport Layer Security (TLS) protocol 1.2
- If you have older versions of Java Application Servers with JDK less than 1.7 (for example, Oracle WebLogic 10.3.6)

For instructions on how to deploy a gateway, see Install a Gateway.

Set the Gateway Variables (Optional)

Set the values for the gateway's host and port.

```
export GW_HOST=<Gateway Host Name>
export GW_PORT=<Gateway Port>
```

If you are using more than one gateway, use the `-additional-gateways` option with the provisioning script.

Download the APM Java Agent Software for Weblogic

1. From the main Oracle Management Cloud menu, navigate to **Administration** and **Agents**.
2. On the Oracle Management Cloud Agents page, click the Action Menu on the top right corner of the page and select **Download Agents**.

   The Agent Software Download page is displayed.
3. From the **Agent Type** dropdown list, select **APM Agent**.
4. Click **APM Java Agent**.
5. Extract the contents of the installer ZIP file.

6. Create a registration key that will be used during the time of installing a new agent. Oracle Application Performance Monitoring Cloud Service verifies this key before accepting any data sent by APM Java Agent deployed on your on-premises hosts. For more information about creating a registration key, see Manage Registration Keys in *Installing and Managing Oracle Management Cloud Agents*.

- Before you install the APM Java Agent, log in to the machine running the application server as a user who installed the application server.
- The application server user should have Read-Write access to the APM Java Agent directories created in the extraction.

Install and Provision the APM Java Agent on Linux

1. Navigate to the directory where you downloaded or copied the APM Java Agent software.
2. Run the provisioning script as per your installation preference:

Installation Preference	Provisioning Script

Basic Installation	`chmod +x ProvisionApmJavaAsAgent.sh`
	`./ProvisionApmJavaAsAgent.sh -d ${DOMAIN_HOME} -no-wallet`

Silent Installation	`chmod +x ProvisionApmJavaAsAgent.sh`
	`./ProvisionApmJavaAsAgent.sh -d ${DOMAIN_HOME} -no-prompt -regkey-file <`*`file name`*`> -no-wallet`

With Gateway
If you are using more than one gateway, use the `-additional-gateways` option.
	`chmod +x ProvisionApmJavaAsAgent.sh`
	`./ProvisionApmJavaAsAgent.sh -d ${DOMAIN_HOME} -no-wallet -gateway-host {gw_host_value} -gateway-port {gw_port_value}`
	`-additional-gateways https://<gw_host_2>:<gw_port_2>,https://<gw_host_3>:<gw_port_3>`

In a proxy environment	`chmod +x ProvisionApmJavaAsAgent.sh`
	`./ProvisionApmJavaAsAgent.sh -d ${DOMAIN_HOME} -no-wallet -ph {http_proxy_host} -pp {http_proxy_port}`
	`-pt {http_proxy_auth_token}`

   - `-d` is the absolute path of the home directory of the Oracle WebLogic Server domain. The APM Java Agent will be installed under this directory.
   - `-ph {http_proxy_host}` (Optional) — the proxy server's host name.

- `-pp {http_proxy_port}` (Optional) — the proxy server's port.
- `-pt {http_proxy_auth_token}`(Optional) — the authorization token that the agent will use if the proxy server requires authentication. This parameter will be passed literally as the proxy authorization header to the proxy server.

  If you are using HTTP Basic authentication, it is recommended that you omit this parameter. For details on HTTP Basic authentication, see Generate a Proxy Token. You can also specify the proxy wallet entry or NTLM credentials token in the following format: `domain/username:password`. For example: `exampledomain/sampleuser:welcome`

The command line displays your tenant name and the value you specified.

3. When prompted, provide the value of the registration key that you've created or downloaded earlier.

   If you are running the provisioning script with the `-no-prompt` option, create a text file containing the value of registration key, and provide the path to the file.

   Example: `ProvisionApmJavaAsAgent.sh -d mydir -no-prompt < regkey.txt`, where `regkey.txt` contains a single line with the registration key.

4. Enable the APM Java Agent. This enables the APM Agent to be invoked when you restart the Oracle WebLogic Server.

   a. Make a backup copy of your `startWebLogic.sh` file:

   ```
   cd $DOMAIN_HOME/bin
   cp startWebLogic.sh startWebLogic.sh.orig
   ```

   b. Edit the `startWebLogic.sh` script.

      i. **If you are installing the APM Java Agent on the WebLogic Administration Server as well as all the Managed Servers:**
      Using a text editor, edit the `startWebLogic.sh` file and add the `-javaagent` option to the set of `JAVA_OPTIONS`. Add the following line after the `setDomainEnv.sh` call:

      ```
      JAVA_OPTIONS=$JAVA_OPTIONS -javaagent:$DOMAIN_HOME/
      apmagent/lib/system/ApmAgentInstrumentation.jar
      ```

      ii. **If you are installing the APM Java Agent on the Managed Servers only:**
      Using a text editor, edit the `startWebLogic.sh` file and add the `-javaagent` option to the set of `JAVA_OPTIONS` within an IF statement similar to that below. Add the following line after the `setDomainEnv.sh` call:

      ```
      if [ "$SERVER_NAME" != "AdminServer" ] ; then
              set
      JAVA_OPTIONS=$JAVA_OPTIONS -javaagent:$DOMAIN_HOME/
      apmagent/lib/system/ApmAgentInstrumentation.jar
      fi
      ```

      Here, `AdminServer` is the name of your Administration Server. Use the exact name with the same space and indentation as above.

5. Restart your Oracle WebLogic application server:

   ```
   % cd $DOMAIN_HOME/bin
   % stopWebLogic.sh
   % cd ..
   % startWebLogic.sh > startup.log
   ```

   > **Note:**
   >
   > Notice that the `$DESTINATION` version of `startWebLogic.sh` is used, even though you edited the `$DESTINATION/bin` version. Invoking the command from one level higher (from `$DESTINATION`) invokes the command from a lower level (from `$DESTINATION/bin`). However, the `stopWebLogic.sh` command will be called from the `$DESTINATION/bin` directory.

6. If you have any managed Oracle WebLogic application servers, restart them:

   ```
   $ cd $DOMAIN_HOME/bin
   $ stopManagedWebLogic.sh {SERVER_NAME} {ADMIN_URL} {USER_NAME}
   {PASSWORD}
   $ startManagedWebLogic.sh {SERVER_NAME} {ADMIN_URL} >
   {SERVER_NAME}.log
   ```

   Run the provisioning script as per your installation preference.

Provision APM Java Agent with a standalone installer
A standalone APM Agent installer is obtained when the Agent zip file is received via email, FTP or similar means (that is, when the agent zip file was not downloaded from an OMC server). To install using the standalone agent installer specify these additional parameters when running the provisioning script:

Option	Description

`-tenant-id`	The Oracle Management Cloud tenant name. You can get this value from the Agent Download page.
	• Script for v1 tenant —
	`sh ProvisionApmJavaAsAgent.sh -d ${DOMAIN_HOME} -tenant-id <tenant> -omc-server-url <omc server url>`
	• Script for v4 tenant —
	`sh ProvisionApmJavaAsAgent.sh -d ${DOMAIN_HOME} -tenant-id <service-tenant> -omc-server-url <omc server url>`

`-omc-server-url`	The URL of the Oracle Management Cloud server. If you are using gateways and have specified `-gateway-host` and `-gateway-port`, you do not have to specify the value for `omc-server-url`.

Provisioning script to run the standalone installer:

Installation Preference	Provisioning Script
Basic Installation	`sh ProvisionApmJavaAsAgent.sh -d ${DOMAIN_HOME} -tenant-id <tenant> -omc-server-url <omc server url> –no-wallet`
With Gateway If you are using more than one gateway, use the -additional-gateways option.	`sh ProvisionApmJavaAsAgent.sh -d ${DOMAIN_HOME} -tenant-id <tenant> -no-wallet -gateway-host  -gateway-port -additional-gateways https://<gw_host_1>:<gw_port_1>,https://<gw_host_2>:<gw_port_2>`
With Gateway and additional Gateways	`sh ProvisionApmJavaAsAgent.sh -d ${DOMAIN_HOME} -tenant-id <tenant> -no-wallet -gateway-host -gateway-port -additional-gateways`
In a proxy environment	`sh ProvisionApmJavaAsAgent.sh -d ${DOMAIN_HOME}  -tenant-id <tenant> -no-wallet -gateway-host -gateway-port -ph {http_proxy_host} -pp {http_proxy_port}`

Generate a Proxy Token

The APM Java Agent can generate a token for HTTP Basic authentication using a username and password instead of a token that is user-generated. Once the user provides a user name and password as property file entries, the APM Java Agent will automatically generate a HTTP Basic authentication token. To get HTTP Basic authentication token generated by the APM Java Agent:

- In the `<agent>/config/AgentHttpBasic.properties` file, specify the following properties:

```
oracle.apmaas.agent.http.proxy.basic.username = myHttpBasicProxyUser
oracle.apmaas.agent.http.proxy.basic.password = myHttpBasicProxyPass
```

> **Note:**
>
> If along with the above properties, an authentication is also specified with the property `oracle.apmaas.agent.http.proxy.token`, it will take precedence over the username and password properties.

APM Java Agent uses the user name and password to generate a HTTP Basic authentication token, and authenticate the user.

Set the NTLM Workstation

Administrators can set the Microsoft Windows NT LAN Manager (NTLM) workstation in case your proxy has this requirement. To set the NTLM workstation:

- Add the following property to the `AgentStartup properties` file:

  `oracle.apmaas.agent.transport.proxy.ntlm.workstation`

Configure NTLM Proxy Authentication in the APM Java Agent

You can configure NTLM proxy authentication in the APM Java Agent to support NTLM authentication.

To configure NTLM proxy authentication:

- In the `<agent>/config/AgentHttpBasic.properties` file, specify the `oracle.apmaas.agent.http.proxy.token` property with a value in this format:

  `domain/username:password`
  Example: `oracle.apmaas.agent.http.proxy.token = testdomain/user1:hello`

On Windows: Requirements and Installation Instructions

Prerequisites for Deploying APM Java Agent on an Oracle WebLogic Server

- Supported versions of Oracle WebLogic Server:
  - Oracle WebLogic Server 12.1.3
  - Oracle WebLogic Server 12.2.1
  - Oracle WebLogic Server 10.3.6, also known as Oracle WebLogic Server 11*g* Release 1

  > **Note:**
  >
  > In the host running Oracle WebLogic Server, you can run the following command from `%WLS_HOME%/server/lib` to check the Oracle WebLogic Server version:
  >
  > ```
  > java -cp weblogic.jar weblogic.version
  > ```

- JDK version 1.7 or above. If you are using JDK 1.6, either use a JDK 1.6 that supports TLS 1.2, or connect to OMC through a Gateway.
- If the JDK version you are running doesn't support TLS 1.2 security protocol, refer to My Oracle Support Doc ID 2703411.1 before proceeding with the APM agent installation.

Other considerations:

- The machine hosting the Oracle WebLogic Server should be able to establish an HTTPS connection either directly or indirectly (using a proxy server or an Oracle Management Cloud gateway) to Oracle Management Cloud. For more information about Oracle Management Cloud gateway, see Install a Gateway.
- The HTTPS connection must use TLS 1.2 security protocol.
- The install user of APM Java Agent should be the same as the Oracle WebLogic Server user.
- The Oracle WebLogic Server user should have read and write permissions to the directories that host the APM Java Agent.

Licensing considerations:

- If you are installing Application Performance Monitoring on the WebLogic Administration Server, then it will also be automatically installed on all the managed servers. One APM Agent will run on each of your managed servers, and this should be considered while calculating the total licensing cost.
- If you prefer to run the APM Agent on one or a few selected managed servers only, then install Application Performance Monitoring on only those managed servers.

Downloading the APM Java Agent Software for Weblogic

1. From the main Oracle Management Cloud menu, navigate to **Administration** and **Agents**.
2. On the Oracle Management Cloud Agents page, click the Action Menu on the top right corner of the page and select **Download Agents**.

   The Agent Software Download page is displayed.
3. From the **Agent Type** dropdown list, select **APM Agent**.
4. Click **APM Java Agent**.
5. Extract the contents of the installer ZIP file.
6. Create a registration key that will be used during the time of installing a new agent. Oracle Application Performance Monitoring Cloud Service verifies this key before accepting any data sent by APM Java Agent deployed on your on-premises hosts. For more information about creating a registration key, see Manage Registration Keys in *Installing and Managing Oracle Management Cloud Agents*.

Installing and Provisioning the APM Java Agent for Windows

1. Navigate to the directory where you downloaded or copied the APM Java Agent software.
2. Run the provisioning script as per your installation preference:

Installation Preference	Provisioning Script

Basic Installation	`set DOMAIN_HOME=<path to domain home>`
	`ProvisionApmJavaAsAgent.cmd /d`
	`%DOMAIN_HOME% /no-wallet`

Installation Preference	Provisioning Script

Silent Installation	`set DOMAIN_HOME=<path to domain home>`
	`ProvisionApmJavaAsAgent.cmd /d %DOMAIN_HOME% /no-prompt /regkey-file <file name> /no-wallet`

With Gateway
If you are using more than one gateway, use the `-additional-gateways` option.
	`set DOMAIN_HOME=<path to domain home>`
	`ProvisionApmJavaAsAgent.cmd /d %DOMAIN_HOME% /no-wallet /gateway-host {gw_host_value} /gateway-port {gw_port_value} /additional-gateways https://<gw_host_1>:<gw_port_1>,https://<gw_host_2>:<gw_port_2>`

In a proxy environment	`set DOMAIN_HOME=<path to domain home>`
	`ProvisionApmJavaAsAgent.cmd /d %DOMAIN_HOME% /no-wallet /ph {http_proxy_host} /pp {http_proxy_port} /pt {http_proxy_auth_token}`

- `-d` is the absolute path of the home directory of the Oracle WebLogic Server domain. The APM Java Agent will be installed under this directory.
- `-ph {http_proxy_host}` (Optional) — the proxy server's host name.
- `-pp {http_proxy_port}` (Optional) — the proxy server's port.
- `-pt {http_proxy_auth_token}`(Optional) — the authorization token that the agent will use if the proxy server requires authentication. This parameter will be passed literally as the proxy authorization header to the proxy server.

The command line displays your tenant name and the value you specified.

3. When prompted, provide the value of the registration key that you've created or downloaded earlier.

   If you are running the provisioning script with the `-no-prompt` option, create a text file containing the value of registration key, and provide the path to the file.

   Example: `ProvisionApmJavaAsAgent.cmd -d mydir -no-prompt < regkey.txt`, where `regkey.txt` contains a single line with the registration key.

4. Enable the APM Java Agent. This enables the APM Agent to be invoked when you restart the Oracle WebLogic Server.

   a. Make a backup copy of your `startWebLogic.cmd` file:

   ```
   cd %DOMAIN_HOME%\bin
   copy startWebLogic.cmd startWebLogic.cmd.orig
   ```

   b. Edit the `startWebLogic.cmd` script.

i. **If you are installing the APM Java Agent on the WebLogic Administration Server as well as all the Managed Servers:**
Using a text editor, edit the `startWebLogic.cmd` file and add the -`javaagent` option to the set of `JAVA_OPTIONS`. Add the following line after the `setDomainEnv.cmd` call:

```
set JAVA_OPTIONS=%JAVA_OPTIONS% -javaagent:%DOMAIN_HOME%
\apmagent\lib\system\ApmAgentInstrumentation.jar
```

ii. **If you are installing the APM Java Agent on the Managed Servers only:**
Using a text editor, edit the `startWebLogic.cmd` file and add the -`javaagent` option to the set of `JAVA_OPTIONS` within an IF statement similar to that below. Add the following line after the `setDomainEnv.cmd` call:

```
if NOT "%SERVER_NAME%"=="AdminServer" (
        set
JAVA_OPTIONS=%JAVA_OPTIONS% -javaagent:%DOMAIN_HOME%
\apmagent\lib\system\ApmAgentInstrumentation.jar
)
```

Here, `AdminServer` is the name of your Administration Server. Use the exact name with the same space and indentation as above.

5. Restart your Oracle WebLogic application server:

```
 % cd %DOMAIN_HOME%\bin
 % stopWebLogic.cmd
 % cd ..
 % startWebLogic.cmd > startup.log
```

> **Note:**
>
> Notice that the `%DESTINATION%` version of `startWebLogic.cmd` is used, even though you edited the `%DESTINATION%/bin` version. Invoking the command from one level higher (from `%DESTINATION%`) invokes the command from a lower level (from `%DESTINATION%/bin`). However, the `stopWebLogic.cmd` command will be called from the `%DESTINATION%/bin` directory.

6. If you have any managed Oracle WebLogic application servers, restart them:

```
% cd %DOMAIN_HOME%\bin
        % stopManagedWebLogic.cmd {SERVER_NAME} {ADMIN_URL}
{USER_NAME} {PASSWORD}
        % startManagedWebLogic.cmd {SERVER_NAME} {ADMIN_URL} >
{SERVER_NAME}.log
```

> **Note:**
>
> If you are running Oracle WebLogic Server as a Microsoft Windows service, add the `-javaagent` flag to the `JAVA_OPTIONS` of your custom registration script, and register your WebLogic Windows service again.

Generating a Proxy Token

The APM Java Agent can generate a token for HTTP Basic authentication using a username and password instead of a token that is user-generated.

Once the user provides a user name and password as property file entries, the APM Java Agent will automatically generate a HTTP Basic authentication token. To get HTTP Basic authentication token generated by the APM Java Agent:

1. In the `<agent>/config/AgentHttpBasic.properties` file, specify the following properties:

   ```
   oracle.apmaas.agent.http.proxy.basic.username = myHttpBasicProxyUser
   oracle.apmaas.agent.http.proxy.basic.password = myHttpBasicProxyPass
   ```

   > **Note:**
   >
   > If along with the above properties, an authentication is also specified with the property `oracle.apmaas.agent.http.proxy.token`, it will take precedence over the username and password properties.

Setting the NTLM Workstation

Administrators can set the Microsoft Windows NT LAN Manager (NTLM) workstation in case your proxy has this requirement. To set the NTLM workstation:

- Add the following property to the `AgentStartup properties` file:

  `oracle.apmaas.agent.transport.proxy.ntlm.workstation`

Configuring NTLM Proxy Authentication in the APM Java Agent

You can configure NTLM proxy authentication in the APM Java Agent to support NTLM authentication.

To configure NTLM proxy authentication:

- In the `AgentHttpBasic.properties` file, specify the `oracle.apmaas.agent.http.proxy.token` property with a value in this format:

  `domain/username:password`
  Example: `oracle.apmaas.agent.http.proxy.token = testdomain/user1:hello`

APM Java Agent Custom Installations

1. Install on Managed Servers through the Administration Console

2. Install when the target server is launched with a custom script

Install on Managed Servers through the Administration Console

If `nodemanager.properties` does not contain `StartScriptName=startWebLogic.cmd` , `StartScriptName=startWebLogic.sh`, or a similar startup script, you will need to add the `-javaagent` flag for managed servers through the WebLogic Server Administration Console.

1. On the WLS Admin Console, navigate to the configuration page for the target managed server. (Environment, Servers and <*server name*>).
2. Click on the Server Start tab. Add the `-javaagent` flag to the *Arguments* text box. If the **Arguments** text box is greyed out, click the **Lock & Edit** button typically in the upper left of the WebLogic Server Administration Console.
   Example: `-javaagent:<full_path>/apmagent/lib/system/ApmAgentInstrumentation.jar`.
3. Click **Save** .
4. If the domain configuration is still locked, click the **Release Configuration** button typically located directly beneath the **Lock & Edit** button.
5. Restart the managed server

Install when the target server is launched with a custom script

When installing the APM Java agent on a server launched with a custom startup script, add the `-javaagent` flag to the command that launches the application you want to monitor.

The custom script usually contains a `Java` executable, a number of Java options, and then either a Main Class (for example, `java $JAVA_OPTIONS Bootstrap`) or a jar containing the Main Class (for example, `java $JAVA_OPTIONS -jar Bootstrap.jar`).

Add the `-javaagent` flag after the `java` executable and before the main class jar. Ensure that the `-javaagent` flag is added as its own standalone flag, and not as part of some other flag's argument. For example, don't add the `-javaagent` flag directly after a standalone `-classpath` option, but add after a `-classpath <value>` combination.

Example:

```
java -classpath $CLASSPATH -javagent:<path_to_agent> -jar Bootstrap.jar
```

To verify if the `javaagent` was successfully added to the application's custom start script, check the command line arguments of the target Java process once it is running.

- **On Linux:**

  ```
  ps -ef | grep java
  ```

  This will list all the Java processes, and you can confirm whether the desired server process has a proper `-javaagent:<path_to_agent>` option
- **On Windows:**
  1. In the Task Manager, Processes tab, select the desired Java process.

2. Right click the process and view the Details tab and check for the commandLine section.
3. Check the CommandLine section, and confirm if the desired java process has a proper `-javaagent:<path_to_agent>` option.

Verify APM Java Agent Installation on Linux

You can verify that the installation of Oracle Application Performance Monitoring Cloud Service is successful by examining the logs and verifying that the user interface displays the Application Server. You can also verify the structure of the installation directory.

Verify that the installation of Oracle Application Performance Monitoring is successful by:

1. Examine the APM Java Agent Logs
2. Use the Oracle Application Performance Monitoring Web Console
3. Verify the APM Java Agent Directory Structure

Examine the APM Java Agent Logs

Examine the log files after installing Oracle Application Performance Monitoring:

1. Verify that the Oracle Application Performance Monitoring log directories and files were created.
   a. After restarting the application server, verify that the APM Java Agent created a log directory for each server it is now monitoring:

      ```
      % cd $DESTINATION/apmagent/logs
      % ls -lF
      ```

      Verify that the following log directory was created: `$DESTINATION/apmagent/logs/<application server name>`
   b. If there are multiple servers in the domain, as each server is restarted, it will be represented by a separate directory under `$DESTINATION/apmagent/logs.`

      For example, if you are monitoring the Application Server, *AdminServer1*, you should see the following entry: `$DESTINATION/apmagent/logs/AdminServer1`
   c. Verify that the correct set of log files were created inside each server log directory:

      ```
      % cd $DESTINATION/apmagent/logs/AdminServer
      % ls -lF *.log
      ```

      Verify that the following set of log files were added to the directory along with other files:
      - `AgentErrors.log`
      - `Agent.log`

- `AgentStartup.log`
- `AgentStatus.log`

If all the expected log directories and the log files were not created, then the Oracle Application Performance Monitoring installation was not successful.

2. Check for errors in the `AgentErrors.log` file. The `AgentErrors.log` file should have a line similar to the following:

```
% more AgentErrors.log
<2015-01-16T12:36:27.27-0800> INFO Exception log is initialized
```

3. In the `AgentStartup.log` file, the message *Agent startup successfully completed* should be seen.
4. Look for agent activity in the `AgentStatus.log` file.

   As the APM Java Agent starts monitoring traffic, it logs short status information in the `AgentStatus.log` file.

   If the traffic and transport counts are more than zero, then that indicates that the APM Java Agent is active, and it is monitoring and reporting data successfully.

Verifying the Installation Using the Oracle Application Performance Monitoring Web Console

Access the Oracle Application Performance Monitoring web user interface from the Oracle Management Cloud home page and verify that the Oracle Application Performance Monitoring installation was successful.

To check for successful Oracle Application Performance Monitoring installation from the web console:

1. Log in to the Oracle Management Cloud home page.
2. In the Oracle Management Cloud Home page, click the **Application Performance Monitoring** tile.

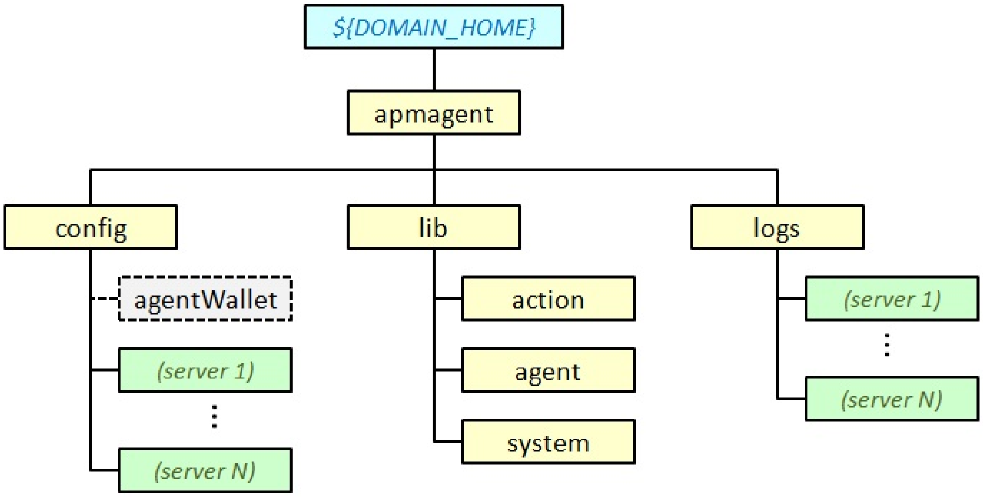
   The Oracle Application Performance Monitoring home page is displayed.
3. Ensure that your user name is displayed in the upper right corner of the home page.
4. To verify that your Application Server was discovered, in the Oracle Application Performance Monitoring home page, click **AppServers**.

   Your application server should be displayed in the AppServers view.
5. Use your applications, and then check for data in the Application Performance Monitoring web console.
   a. Use the application that you want to monitor, and make multiple transactions.
   b. In the Oracle Application Performance Monitoring home page, click the Time Selector drop-down list and select **Last Hour**.
   c. See the Top 5 Server Requests or click the number above **Server Requests** to see the Server Requests view.
   d. If you are on the Enterprise Edition, click **Pages** to see the Pages view.

Ensure that the operations you performed on the application are reflected in the Server Requests or the Pages view.

Verify the APM Java Agent Directory Structure

After installing and provisioning APM Java Agent, you can find the following directory structure on your WebLogic Managed Server.



Directory	Sub-directory	Description
${DOMAIN_HOME}	—	The home directory of the WebLogic Server domain, where the APM Java Agent is installed.
apmagent	—	The root of the APM Java Agent's installation directory. No files are stored directly in this directory. Note that there will be exactly one APM Java Agent installation directory for this domain (on this host) regardless of how many WebLogic Servers are being monitored.
config/	—	All of the APM Java Agent's domain-level configuration files are stored directly in this directory.
—	agentWallet	If this domain is using an Oracle Wallet to hold the APM Java Agent's Authorization Token (acting as the APM Java Agent's *Credential Store*), then this directory will exist, and will hold the `cwallet.sso` file which is the wallet.
—	(server 1) ...(server N)	For each server in the domain that is being monitored, a configuration directory will be created when the APM Java Agent first discovers that server. Configuration data that can be modified on a server-by-server basis (as opposed to that for the entire domain), will be found here.
lib/	—	All of the APM Java Agent's `.jar` files will be found under the lib directory.
—	action agent system	The three sub-directories under the lib directory.

Directory	Sub-directory	Description
logs/	—	The APM Java Agent's log files will be stored in one of this directory's sub-directories. No files will be found directly in this directory.
—	(server 1) ...(server N)	For each server in the domain being monitored, a log directory will be created when the APM Java Agent first discovers that server. All log files pertaining to a particular server will be stored in these sub-directories.

3

Install and Configure APM Java Agents on Websphere Application Server

Here are the requirements and instructions to install APM Java Agent on a Websphere Application Server.

Websphere: Requirements and Installation Instructions

Prerequisites for Deploying APM Java Agent on a WebSphere Application Server

- Supported versions:
  - WebSphere Application Server 8.5 and 9.0
- JDK version 1.7 or above. If you are using JDK 1.6, either use a JDK 1.6 that supports TLS 1.2 security protocol, or connect to OMC through a Gateway.
- If the JDK version you are running doesn't support TLS 1.2 security protocol, refer to My Oracle Support Doc ID 2703411.1 before proceeding with the APM agent installation.
- **Other considerations:**
  - The machine hosting the WebSphere Application Server should be able to establish an HTTPS connection either directly or indirectly (using a proxy server or an Oracle Management Cloud gateway) to Oracle Management Cloud. For more information about Oracle Management Cloud gateway, see Install a Gateway.
  - The HTTPS connection must use TLS 1.2 security protocol.
  - The install user of APM Java Agent should be the same as the WebSphere Application Server user.
  - The WebSphere Application Server user should have read and write permissions to the directories that host the APM Java Agent, as well as the WebSphere Application Server Home.

Set the `WAS_HOME` Variable

Set the `WAS_HOME` variable to point to the WebSphere Server domain directory.

- If you're using a Bash shell:

  ```
  export WAS_HOME=<WebSphere Server Domain>
  ```

- If you're using a C shell:

  ```
  setenv WAS_HOME "<WebSphere Server Domain>"
  ```

Deploy a Gateway (Optional)

Gateway is not a mandatory component while deploying Oracle Application Performance Monitoring; you can use a gateway in the following scenarios:

- If you have an application server that does not support Transport Layer Security (TLS) protocol 1.2
- If you have older versions of Java Application Servers with JDK less than 1.7 (for example, Oracle WebLogic 10.3.6)

For instructions on how to deploy a gateway, see Install a Gateway.

Set the Gateway Variables (Optional)

Set the values for Gateway host and port.

- If you're using a Bash shell:

  ```
  export GW_HOST=<Gateway Host Name>
  export GW_PORT=<Gateway Port>
  ```

- If you're using a C shell:

  ```
  setenv GW_HOST "<Gateway Host Name>"
  setenv GW_PORT "<Gateway Port>"
  ```

If you are using more than one gateway, use the `-additional-gateways`option with the provisioning script.

Download the APM Java Agent Software for Websphere

1. On the Oracle Management Cloud home page, click the Oracle Management Cloud Navigation icon on the top-left corner to view the Management Cloud navigation pane.
2. Select **Administration** and **Agents**.
3. On the Oracle Management Cloud Agents page, click the Action Menu on the top right corner of the page and select **Download Agents**.

   The Agent Software Download page is displayed.
4. From the **Agent Type** drop-down list, select **APM Agent**.
5. Click **APM Java Agent**.
6. Extract the contents of the installer ZIP file.
7. Create a registration key that will be used during the time of installing a new agent. Oracle Application Performance Monitoring Cloud Service verifies this key before accepting any data sent by APM Java Agent deployed on your on-premises hosts. For more information about creating a registration key, see Manage Registration Keys.

Install and Provision APM Java Agent on WebSphere

- Before you install the APM Java Agent, log in to the machine running the application server as the user identity your WebSphere runs as.

- The application server user should have Read-Write access to the APM Java Agent directories.

To install and provision the APM Java Agent:

1. Navigate to the directory where you downloaded the APM Java Agent software.
2. Run the provisioning script as per your installation preference:

Installation Preference	Provisioning Script
Windows Basic Installation	`ProvisionApmJavaAsAgent.cmd /d ${WAS_HOME} /no-wallet` For example: `ProvisionApmJavaAsAgent.cmd /d "C:\ibm\WebSphere\AppServer" /no-wallet`

Linux Basic Installation	`chmod +x ProvisionApmJavaAsAgent.sh`
	`./ProvisionApmJavaAsAgent.sh -d ${WAS_HOME} -no-wallet`

Silent Installation	`chmod +x ProvisionApmJavaAsAgent.sh`
	`./ProvisionApmJavaAsAgent.sh -d ${WAS_HOME}-no-prompt -no-wallet`

With Gateway in a Linux Environment
If you are using more than one gateway, use the `-additional-gateways` option.
	`chmod +x ProvisionApmJavaAsAgent.sh`
	`./ProvisionApmJavaAsAgent.sh -d ${WAS_HOME} -no-wallet gateway-host {gw_host_value}, -gateway-port {gw_port_value}`
	`-additional-gateways https://<gw_host_1>:<gw_port_1>,https://<gw_host_2>:<gw_port_2>`

With Gateway in a Windows Environment
If you are using more than one gateway, use the `-additional-gateways` option.
	`ProvisionApmJavaAsAgent.cmd /d ${WAS_HOME} /no-wallet gateway-host {gw_host_value} /gateway-port {gw_port_value}`
	`/additional-gateways https://<gw_host_1>:<gw_port_1>,https://<gw_host_2>:<gw_port_2>`

Installation Preference	Provisioning Script

In a Proxy Environment	`chmod +x ProvisionApmJavaAsAgent.sh`
	`./ProvisionApmJavaAsAgent.sh -d ${WAS_HOME} -no-wallet -ph {http_proxy_host} -pp {http_proxy_port} -pt`
	`{http_proxy_auth_token}`

- `-d` is the absolute path of the home directory of your WebSphere installation. The APM Java Agent will be installed under this directory.
- `-ph {http_proxy_host}` (Optional) — the proxy server's host name.
- `-pp {http_proxy_port}` (Optional) — the proxy server's port.
- `-pt {http_proxy_auth_token}`(Optional) — the authorization token that the agent will use if the proxy server requires authentication. This parameter will be passed literally as the proxy authorization header to the proxy server.

  If you are using HTTP Basic authentication, it is recommended that you omit this parameter. For details on HTTP Basic authentication, see Generate Proxy Token. You can also specify the proxy wallet entry or NTLM credentials token in the following format: `domain/username:password`. For example: `exampledomain/sampleuser:welcome`

The command line displays your tenant name and the value you specified.

3. When prompted, provide the value of the registration key that you've created or downloaded earlier.

   If you are running the provisioning script with the `-no-prompt` option, create a text file containing the value of registration key, and provide the path to the file. Example: `./ProvisionApmJavaAsAgent.sh -d mydir -no-prompt < regkey.txt` where `regkey.txt` contains a single line with the registration key.

4. Review the values and confirm. To change the values, enter `n` and run the script again with new values.
5. To proceed with the installation, enter `y`.

   The APM Java Agent is installed.

Provision APM Java Agent with Offline Installer

You can install and provision the APM Java Agent from an offline installer that you have received over email.

To install from the emailed ZIP, the provisioning script needs to download the configuration file from Oracle Management Cloud. Specify these additional options while running the provisioning script:

Option	Description
`-tenant-id`	The Oracle Management Cloud tenant name.
`-omc-server-url`	The URL of the Oracle Management Cloud server. Example: `https://omchost:port`

Modify the Startup Script of Your WebSphere Server

To modify the `server.policy` startup script of your WebSphere server:

1. From your WebSphere Admin console, click the **Servers** tab and select the server on which you want to provision the APM Java Agent.
2. Expand **Java and Process Management** tab and select **Process Definition**.
3. Under **Additional Properties** tab, select **Java Virtual Machine**.
4. In the **Generic JVM arguments** field, add the following line to `–javaagent` and `-Dws.ext.dirs` flags:

   ```
   -javaagent:\$WAS_HOME/apmagent/lib/system/ApmAgentInstrumentation.jar
   -Dws.ext.dirs=\$WAS_HOME/apmagent/lib/agent/ApmEumFilter.jar
   ```

5. Make a copy of your `server.policy` file:

   ```
   % cd $WAS_HOME/properties
   % cp server.policy server.policy.orig
   ```

6. Using a text editor, edit the original `server.policy` file:

   ```
   grant codeBase "file:$WAS_HOME/apmagent/-"
   {
   permission java.security.AllPermission;
   };
   ```

7. From your WebSphere administration console, stop and start the WebSphere server. You can also use the command line:

   ```
   cd $WAS_HOME/bin
   ./stopServer.sh <servername>
   ./startServer.sh <servername>
   ```

Verify APM Java Agent Installation

See Verify APM Java Agent Installation.

4 Install and Configure APM Java Agent on Apache Tomcat

Here are the requirements and instructions to install APM Java Agent on Apache Tomcat.

- Apache Tomcat on Linux: Requirements and Installation Instructions
- Apache Tomcat On Windows: Requirements and Installation Instructions
- Configure the APM Agent as a Windows Service

Apache Tomcat on Linux: Requirements and Installation Instructions

Prerequisites for Deploying APM Java Agent on an Apache Tomcat Server

- Supported versions of Apache Tomcat Server: 6, 7, 8, 8.5, 9 and TomEE.
- JDK version 1.7 or above. If you are using JDK 1.6, either use a JDK 1.6 that supports TLS 1.2 security protocol, or connect to OMC through a Gateway.
- If the JDK version you are running doesn't support TLS 1.2 security protocol, refer to My Oracle Support Doc ID 2703411.1 before proceeding with the APM agent installation.
- **Other considerations:**
  - The machine hosting the Apache Tomcat Server should be able to establish an HTTPS connection either directly or indirectly (using a proxy server or an Oracle Management Cloud gateway) to Oracle Management Cloud. For more information about Oracle Management Cloud gateway, see Install a Gateway.
  - The HTTPS connection must use TLS 1.2 security protocol.
  - The install user of APM Java Agent should be the same as the Apache Tomcat user.
  - The Apache Tomcat Server user should have read and write permissions to the directories that host the APM Java Agent, as well as the `CATALINA_BASE` APM Java Agent logging and config directories if different from `CATALINA_HOME`.
  - If you are installing the APM Java Agent on minor Tomcat version 6.0.20 or earlier, please add the following command line option to enable local monitoring:

    ```
    -Dcom.sun.management.jmxremote
    ```

Set the `DESTINATION` Variable

Set the `CATALINA_HOME` variable to point to the Tomcat destination directory.

- If you're using a Bash shell:

  ```
  export CATALINA_HOME=<Tomcat destination directory>
  ```

- If you're using a C shell:

  ```
  setenv CATALINA_HOME "<Tomcat destination directory>"
  ```

Deploy a Gateway (Optional)

Gateway is not a mandatory component while deploying Oracle Application Performance Monitoring; you can use a gateway in the following scenarios:

- If you have an application server that does not support Transport Layer Security (TLS) protocol 1.2
- If you have older versions of Java Application Servers with JDK less than 1.7 (for example, Oracle WebLogic 10.3.6)

For instructions on how to deploy a gateway, see Install a Gateway.

Set the Gateway Variables (Optional)

Set the values for Gateway host and port.

- If you're using a Bash shell:

  ```
  export GW_HOST=<Gateway Host Name>
  export GW_PORT=<Gateway Port>
  ```

- If you're using a C shell:

  ```
  setenv GW_HOST "<Gateway Host Name>"
  setenv GW_PORT "<Gateway Port>"
  ```

If you are using more than one gateway, use the `-additional-gateways` option with the provisioning script.

Download the APM Java Agent Software for Apache Tomcat

1. From the main Oracle Management Cloud menu, navigate to **Administration** and **Agents**.
2. On the Oracle Management Cloud Agents page, click the Action Menu on the top right corner of the page and select **Download Agents**.

   The Agent Software Download page is displayed.
3. From the **Agent Type** dropdown list, select **APM Agent**.
4. Click **APM Java Agent**.
5. Extract the contents of the installer ZIP file.

Install and Provision APM Java Agent on Apache Tomcat

- Before you install the APM Java Agent, log in to the machine running the application server as a user who installed the application server.
- The application server user should have Read-Write access to the APM Java Agent directories.

To install and provision the APM Java Agent:

1. Navigate to the directory where you downloaded the APM Java Agent software.
2. Run the provisioning script as per your installation preference:

Installation Preference	Provisioning Script

Basic Installation	`cd ${STAGE_DIR}`
	`chmod +x ProvisionApmJavaAsAgent.sh`
	`./ProvisionApmJavaAsAgent.sh -d ${CATALINA_HOME}`

Silent Installation	`chmod +x ProvisionApmJavaAsAgent.sh`
	`./ProvisionApmJavaAsAgent.sh -d ${CATALINA_HOME} -no-prompt`

With Gateway
If you are using more than one gateway, use the `-additional-gateways` option.
	`chmod +x ProvisionApmJavaAsAgent.sh`
	`./ProvisionApmJavaAsAgent.sh -d ${CATALINA_HOME} -gateway-host {gw_host_value} -gateway-port {gw_port_value} -additional-gateways https://<gw_host_1>:<gw_port_1>,https://<gw_host_2>:<gw_port_2>`

In a Proxy Environment	`chmod +x ProvisionApmJavaAsAgent.sh`
	`./ProvisionApmJavaAsAgent.sh -d ${CATALINA_HOME} -ph {http_proxy_host} -pp {http_proxy_port} -pt {http_proxy_auth_token}`

- `-d` is the absolute path of the home directory of the WebLogic Server domain. The APM Java Agent will be installed under this directory.
- `-ph {http_proxy_host}` (Optional) — the proxy server's host name.
- `-pp {http_proxy_port}` (Optional) — the proxy server's port.
- `-pt {http_proxy_auth_token}`(Optional) — the authorization token that the agent will use if the proxy server requires authentication. This parameter will be passed literally as the proxy authorization header to the proxy server.

The command line displays your tenant name and the value you specified.

3. When prompted, provide the value of the registration key that you've created or downloaded earlier.

   If you are running the provisioning script with the `-no-prompt` option, create a text file containing the value of registration key, and provide the path to the file. Example: `./ProvisionApmJavaAsAgent.sh -d mydir -no-prompt < regkey.txt` where `regkey.txt` contains a single line with the registration key.

4. Review the values and confirm. To change the values, enter `n` and run the script again with new values.

5. To proceed with the installation, enter `y`.

   The APM Java Agent is installed.

`ProvisionApmJavaAsAgent.sh`Syntax for Using the Installation Script

Install APM Java Agent on a Microservice
When running the APM Java Agent provisioning script on an application or microservice using Spring Boot or similar embedded Tomcat framework, the installation is very similar to standalone Tomcat. First, if your microservice does not have a standard server home (DOMAIN_HOME, CATALINA_HOME etc.), you can simply choose an empty directory for it. Once it's provisioned, add:

```
-javaagent:<path-to-agent>/apmagent/lib/system/
ApmAgentInstrumentation.jar
```

to the startup command line or script for your application or microservice.

Example:

```
java -javaagent:/u01/apmagent/lib/system/ApmAgentInstrumentation.jar -
jar my-microservice-1.0-SNAPSHOT.jar
```

Provisioning APM Java Agent with Offline Installer
You can install and provision the APM Java Agent from an offline installer that you have received over email.

To install from the emailed ZIP, the provisioning script needs to download the configuration file from Oracle Management Cloud. Specify these additional options while running the provisioning script:

Option	Description
`-tenant-id`	The Oracle Management Cloud tenant name.
`-omc-server-url`	The URL of the Oracle Management Cloud server. Example: `https://omchost:port`

Modify the Startup Script of Your Apache Tomcat Server

To modify the `catalina.sh` script of your Tomcat server home:

1. Make a copy of your `catalina.sh` file:

   ```
   % cd $CATALINA_HOME/bin
   % cp catalina.sh catalina.sh.orig
   ```

2. Using a text editor, edit the original `catalina.sh` file and add `-javaagent` to `CATALINA_OPTS`. Make the change outside of any `if` statements or code blocks that may not be executed during server startup. This will ensure the `-javaagent` flag is always added to the server startup options.

   ```
   CATALINA_OPTS="$CATALINA_OPTS -javaagent:${CATALINA_HOME}/
   apmagent/lib/system/ApmAgentInstrumentation.jar -
   Datlassian.org.osgi.framework.bootdelegation=oracle.apmaas.*,sun.*,o
   rg.apache.xerces,org.apache.xerces.*,org.apache.naming,org.apache.na
   ming.*,org.apache.catalina,org.apache.catalina.*,org.apache.tomcat.u
   til.*"
   ```

   The `-Datlassian.org.osgi.framework.bootdelgation` option is required if you are running an Atlassian Jira application.

3. Restart the Tomcat Servers:

   ```
   % cd $CATALINA_HOME/bin
   % ./shutdown.sh
   % ./startup.sh
   ```

Verify APM Java Agent Installation

See Verify APM Java Agent Installation.

Apache Tomcat On Windows: Requirements and Installation Instructions

Download the APM Java Agent Software for Apache Tomcat

1. From the main Oracle Management Cloud menu, navigate to **Administration** and **Agents**.
2. On the Oracle Management Cloud Agents page, click the Action Menu on the top right corner of the page and select **Download Agents**.

   The Agent Software Download page is displayed.
3. From the **Agent Type** dropdown list, select **APM Agent**.
4. Click **APM Java Agent**.
5. Extract the contents of the installer ZIP file.

Install and Provision APM Java Agent on Apache Tomcat on Windows

- Before you install the APM Java Agent, log in to the machine running the application server as a user who installed the application server.
- The application server user should have Read-Write access to the APM Java Agent directories.

To install and provision the APM Java Agent:

1. Navigate to the directory where you downloaded the APM Java Agent software.
2. Run the provisioning script as per your installation preference:

Installation Preference	Provisioning Script
Basic Installation	`ProvisionApmJavaAsAgent.cmd /d %CATALINA_HOME% -no-wallet`
Silent Installation	Download the registration key from the console and provide the value: `ProvisionApmJavaAsAgent.cmd /d %CATALINA_HOME%-no-prompt -regkey-file key.txt`
With Gateway If you are using more than one gateway, use the `-additional-gateways` option.	`ProvisionApmJavaAsAgent.cmd /d %CATALINA_HOME% -no-wallet -gateway-host {gw_host_value}, -gateway-port {gw_port_value} -additional-gateways https://<gw_host_1>:<gw_port_1>,https://<gw_host_2>:<gw_port_2>`
In a Proxy Environment	`ProvisionApmJavaAsAgent.cmd /d %CATALINA_HOME% -no-wallet -ph {http_proxy_host} -pp {http_proxy_port} -pt {http_proxy_auth_token}`

- `-d` is the absolute path of the home directory of Apache Tomcat. The APM Java Agent will be installed under this directory.
- `-ph {http_proxy_host}` (Optional) — the proxy server's host name.
- `-pp {http_proxy_port}` (Optional) — the proxy server's port.
- `-pt {http_proxy_auth_token}`(Optional) — the authorization token that the agent will use if the proxy server requires authentication. This parameter will be passed literally as the proxy authorization header to the proxy server.

The command line displays your tenant name and the value you specified.

3. When prompted, provide the value of the registration key that you've created or downloaded earlier.

   If you are running the provisioning script with the `-no-prompt` option, create a text file containing the value of registration key, and provide the path to the file. Example: `ProvisionApmJavaAsAgent.cmd /d mydir -no-prompt < regkey.txt` where `regkey.txt` contains a single line with the registration key.

4. Review the values and confirm. To change the values, enter `n` and run the script again with new values.
5. To proceed with the installation, enter `y`.

The APM Java Agent is installed.

Modify the Startup Script of Your Apache Tomcat Server

To modify the `catalina.bat` script of your Tomcat server home:

1. Make a copy of your `catalina.bat` file:

   ```
   cd %CATALINA_HOME%\bin
   % cp catalina.bat catalina.bat.orig
   ```

2. Using a text editor, edit the original `catalina.bat` file and add the `-javaagent` to `CATALINA_OPTS`. Make the change outside of any `if` statements or code blocks that may not be executed during server startup. This will ensure the `-javaagent` flag is always added to the server startup options.

   ```
   set CATALINA_OPTS=%CATALINA_OPTS%
         -javaagent:"%CATALINA_HOME%
   \apmagent\lib\system\ApmAgentInstrumentation.jar"
   ```

   The `-Datlassian.org.osgi.framework.bootdelgation` option is required if you are running an Atlassian Jira application.

3. Restart the Tomcat Servers:

   ```
   cd %CATALINA_HOME%\bin
   shutdown.bat
   startup.bat
   ```

Verify APM Java Agent Installation

See Verify APM Java Agent Installation.

Configure the APM Agent as a Windows Service

Before you can manage your application, you must configure the APM Agent in Apache Tomcat as a Windows service.

Set the `JvmArgs` variable depending on your Apache Tomcat version:

- **Apache Tomcat 9**
  1. Stop the Apache server:

     ```
     C:\tomcat\apache-tomcat-9.0.2-windows-x64\apache-tomcat-9.0.2/
     bin>shutdown
     ```

  2. Run the following command:

     ```
     set JvmArgs=javaagent:<path-to-APM>/apmagent/lib/system/
     ApmAgentInstrumentation.jar
     ```

     – **Example:**

set **CATALINA_HOME**=" C:\tomcat\apache-tomcat-9.0.2-windows-x64\apache-tomcat-9.0.2"

```
set JvmArgs=javaagent:%CATALINA_HOME%/apmagent/lib/system/ApmAgentInstrumentation.jar
```

3. Start your Apache server:

   ```
   C:\tomcat\apache-tomcat-9.0.2-windows-x64\apache-tomcat-9.0.2/bin>startup
   ```

- **Apache Tomcat 8.5, 8, 7, 6**

  1. Change directory to the Apache bin folder:

     ```
     cd C:\tomcat\apache-tomcat-8.5.27-windows-x64\apache-tomcat-8.5.27\bin
     ```

     Stop the Apache server:

     ```
     C:\tomcat\apache-tomcat-8.5.27-windows-x64\apache-tomcat-8.5.27/bin>shutdown
     ```

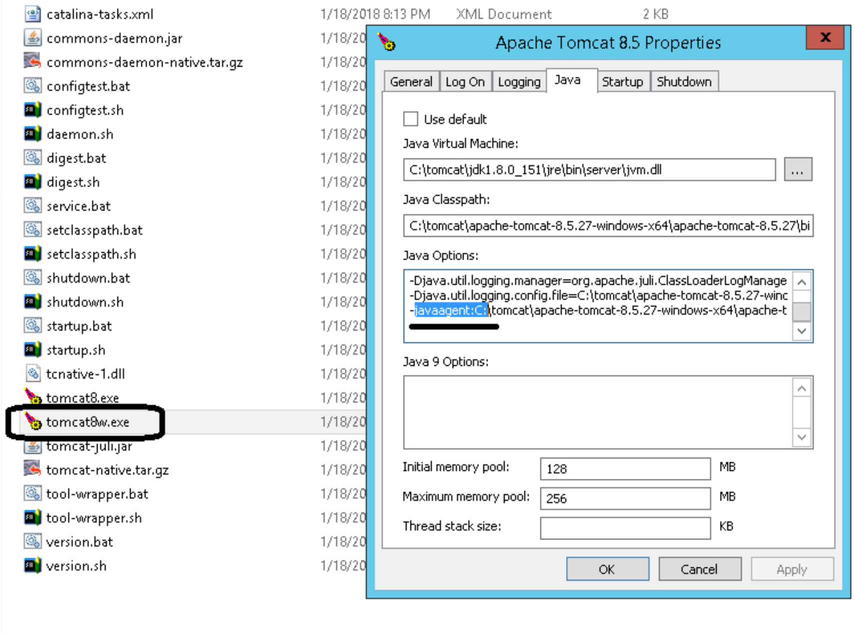
  2. Double click `tomcat<version_number>w.exe` or run it from the command line.

     For example, if you have Tomcat version 8, double click or run `tomcat8w.exe`.

     > **Note:**
     >
     > If you are installing APM Agent on a TomEE server, select `TomEE.exe` instead. If you have trouble using TomEE.exe, see this troubleshooting tip, Unable to open TomEE service during installation.

  3. When the **Properties** window pops up, click the **Java** tab and add the following line to the **Java Options**:

     ```
     -javaagent:C:\tomcat\apache-tomcat-8.5.27-windows-x64\apache-tomcat-8.5.27\apmagent\lib\system\ApmAgentInstrumentation.jar
     ```



catalina-tasks.xml 1/18/2018 8:13 PM XML Document 2 KB
commons-daemon.jar 1/18/20
commons-daemon-native.tar.gz 1/18/20
configtest.bat 1/18/20
configtest.sh 1/18/20
daemon.sh 1/18/20
digest.bat 1/18/20
digest.sh 1/18/20
service.bat 1/18/20
setclasspath.bat 1/18/20
setclasspath.sh 1/18/20
shutdown.bat 1/18/20
shutdown.sh 1/18/20
startup.bat 1/18/20
startup.sh 1/18/20
tcnative-1.dll 1/18/20
tomcat8.exe 1/18/20
tomcat8w.exe 1/18/20
tomcat-juli.jar 1/18/20
tomcat-native.tar.gz 1/18/20
tool-wrapper.bat 1/18/20
tool-wrapper.sh 1/18/20
version.bat 1/18/20
version.sh 1/18/20

Apache Tomcat 8.5 Properties

□ Use default

Java Virtual Machine: C:\tomcat\jdk1.8.0_151\jre\bin\server\jvm.dll

Java Classpath: C:\tomcat\apache-tomcat-8.5.27-windows-x64\apache-tomcat-8.5.27\bi

Java Options:

```
-Djava.util.logging.manager=org.apache.juli.ClassLoaderLogManage
-Djava.util.logging.config.file=C:\tomcat\apache-tomcat-8.5.27-winc
-javaagent:C:\tomcat\apache-tomcat-8.5.27-windows-x64\apache-t
```

Java 9 Options:

Initial memory pool: 128 MB

Maximum memory pool: 256 MB

Thread stack size: KB

4. Start your Apache server:

```
C:\tomcat\apache-tomcat-8.5.27-windows-x64\apache-tomcat-8.5.27/
bin>startup
```

5 Install and Configure APM Java Agent on JBoss

Here are the requirements and instructions to install APM Java Agent on JBoss.

JBoss: Requirements and Installation Instructions

Prerequisites for Deploying APM Java Agent on a JBoss Server

- Supported versions:
  - JBoss EAP 6.1.1+
  - Wildfly 9.0.2
- JDK version 1.7 or above. If you are using JDK 1.6, either use a JDK 1.6 that supports TLS 1.2 security protocol, or connect to OMC through a Gateway.
- If the JDK version you are running doesn't support TLS 1.2 security protocol, refer to My Oracle Support Doc ID 2703411.1 before proceeding with the APM agent installation.
- **Other considerations:**
  - The machine hosting the JBoss Server should be able to establish an HTTPS connection either directly or indirectly (using a proxy server or an Oracle Management Cloud gateway) to Oracle Management Cloud. For more information about Oracle Management Cloud gateway, see Install a Gateway.
  - The HTTPS connection must use TLS 1.2 security protocol.
  - The install user of APM Java Agent should be the same as the JBoss user.
  - The JBoss Server user should have read and write permissions to the directories that host the APM Java Agent, as well as the JBOSS Home.

Set the `JBOSS_HOME` Variable

Set the `JBOSS_HOME` variable to point to the JBoss home directory.

- If you're using a Bash shell:

  ```
  export JBOSS_HOME=<JBoss home directory>
  ```

- If you're using a C shell:

  ```
  setenv JBOSS_HOME "<JBoss home directory>"
  ```

Deploy a Gateway (Optional)

Gateway is not a mandatory component while deploying Oracle Application Performance Monitoring; you can use a gateway in the following scenarios:

- If you have an application server that does not support Transport Layer Security (TLS) protocol 1.2
- If you have older versions of Java Application Servers with JDK less than 1.7 (for example, Oracle WebLogic 10.3.6)

For instructions on how to deploy a gateway, see Install a Gateway.

Set the Gateway Variables (Optional)

Set the values for Gateway host and port.

- If you're using a Bash shell:

  ```
  export GW_HOST=<Gateway Host Name>
  export GW_PORT=<Gateway Port>
  ```

- If you're using a C shell:

  ```
  setenv GW_HOST "<Gateway Host Name>"
  setenv GW_PORT "<Gateway Port>"
  ```

If you are using more than one gateway, use the `-additional-gateways` option with the provisioning script.

Download the APM Java Agent Software for JBoss

1. On the Oracle Management Cloud home page, click the Oracle Management Cloud Navigation icon on the top-left corner to view the Management Cloud navigation pane.
2. Select **Administration** and **Agents**.
3. On the Oracle Management Cloud Agents page, click the Action Menu on the top right corner of the page and select **Download Agents**.

   The Agent Software Download page is displayed.
4. From the **Agent Type** dropdown list, select **APM Agent**.
5. Click **APM Java Agent**.
6. Extract the contents of the installer ZIP file.
7. Create a registration key that will be used during the time of installing a new agent. Oracle Application Performance Monitoring Cloud Service verifies this key before accepting any data sent by APM Java Agent deployed on your on-premises hosts. For more information about creating a registration key, see Manage Registration Keys in *Installing and Managing Oracle Management Cloud Agents*.

Install and Provision APM Java Agent on JBoss

- Before you install the APM Java Agent, log in to the machine running the application server as a user who installed the application server.
- The application server user should have Read-Write access to the APM Java Agent directories.

To install and provision the APM Java Agent:

1. Navigate to the directory where you downloaded the APM Java Agent software.
2. Run the provisioning script as per your installation preference:

Installation Preference	Provisioning Script

Basic Installation	`chmod +x ProvisionApmJavaAsAgent.sh`
	`./ProvisionApmJavaAsAgent.sh -d ${JBOSS_HOME} -no-wallet`

Silent Installation	`chmod +x ProvisionApmJavaAsAgent.sh`
	`./ProvisionApmJavaAsAgent.sh -d ${JBOSS_HOME} -no-prompt`

With Gateway
If you are using more than one gateway, use the `-additional-gateways` option.
	`chmod +x ProvisionApmJavaAsAgent.sh`
	`./ProvisionApmJavaAsAgent.sh -d ${JBOSS_HOME} -no-wallet -gateway-host {gw_host_value}, -gateway-port {gw_port_value}, -additional-gateways https://<gw_host_1>:<gw_port_1>,https://<gw_host_2>:<gw_port_2>`

In a Proxy Environment	`chmod +x ProvisionApmJavaAsAgent.sh`
	`./ProvisionApmJavaAsAgent.sh -d ${JBOSS_HOME} -no-wallet -ph {http_proxy_host} -pp {http_proxy_port} -pt {http_proxy_auth_token}`

- `-d` is the absolute path of the home directory of your JBoss installation. The APM Java Agent will be installed under this directory.
- `-ph {http_proxy_host}` (Optional) — the proxy server's host name.
- `-pp {http_proxy_port}` (Optional) — the proxy server's port.
- `-pt {http_proxy_auth_token}`(Optional) — the authorization token that the agent will use if the proxy server requires authentication. This parameter will be passed literally as the proxy authorization header to the proxy server.

  If you are using HTTP Basic authentication, it is recommended that you omit this parameter. For details on HTTP Basic authentication, see Generate Proxy Token. You can also specify the proxy wallet entry or NTLM credentials token in the following format: `domain/username:password`. For example: `exampledomain/sampleuser:welcome`

The command line displays your tenant name and the value you specified.

3. When prompted, provide the value of the registration key that you've created or downloaded earlier.

   If you are running the provisioning script with the `-no-prompt` option, create a text file containing the value of registration key, and provide the path to the file. Example: `./ProvisionApmJavaAsAgent.sh -d mydir -no-prompt < regkey.txt` where `regkey.txt` contains a single line with the registration key.

4. Review the values and confirm. To change the values, enter `n` and run the script again with new values.
5. To proceed with the installation, enter `y`.

   The APM Java Agent is installed.

Provisioning APM Java Agent with Offline Installer
You can install and provision the APM Java Agent from an offline installer that you have received over email.

To install from the emailed ZIP, the provisioning script needs to download the configuration file from Oracle Management Cloud. Specify these additional options while running the provisioning script:

Option	Description
`-tenant-id`	The Oracle Management Cloud tenant name.
`-omc-server-url`	The URL of the Oracle Management Cloud server. Example: `https://omchost:port`

Modify the Startup Script of Your JBoss Server

Modify the startup script of your application server to invoke Oracle Application Performance Monitoring, when you start your JBoss/WildFly server.

To modify the `standalone.conf` script of your JBoss/WildFly server:

1. Make a copy of your `standalone.conf` file:

```
% cd $JBOSS_HOME/bin
% cp standalone.conf standalone.conf.orig
```

2. Move to the end of the `standalone.conf` file, and add the following Java option:

   ```
    JAVA_OPTS="$JAVA_OPTS
   -
   Djboss.modules.system.pkgs=org.jboss.byteman,oracle.security.pki,ora
   cle.apmaas.agent,oracle.apmaas.repackaged"
   JAVA_OPTS="$JAVA_OPTS -javaagent:$JBOSS_HOME/apmagent/lib/system/
   ApmAgentInstrumentation.jar"
   ```

3. Stop and restart the JBoss Server:

   ```
   % cd $JBOSS_HOME/bin
   % ./jboss-cli.sh -c :shutdown
   % nohup ./standalone.sh -b 0.0.0.0&> startup.log &
   ```

   You can check the entries present in the *`$JBOSS_HOME`*`/startup.log` file to verify that the JBoss Server has started.

4. If you have any managed JBoss application servers, stop and restart them.

Install and Provision APM Java Agent on JBoss on Microsoft Windows

You can install and provision the APM Java Agent on your JBoss domain installed on a Microsoft Windows machine.

Before you install:

- Log in to the machine running the application server, as the same user who installed the application server.
- Ensure that the application server user has Read-Write access to the APM Java Agent directories.

To install and provision APM Java Agent on JBoss Server on Microsoft Windows:

1. Run the following command to provision the APM Java Agent:

   ```
   set the DOMAIN HOME : set JBOSS_HOME=<JBoss_Home Path>
   ProvisionApmJavaAsAgent.cmd /d %JBOSS_HOME% /no-wallet
   ```

2. Provision the APM Java agent. This enables the APM Agent to be invoked when you restart the JBoss Server.

   a. Make a backup copy of your `standalone.conf.bat` file:

      ```
      %cd %JBOSS_HOME%\bin
      %cp standalone.conf.bat standalone.conf.bat.orig
      ```

3. Edit the `standalone.conf.bat` file, add the following lines at the end of the file:

   ```
   set "JAVA_OPTS=%JAVA_OPTS% -
   Djboss.modules.system.pkgs=org.jboss.byteman,oracle.security.pki,ora
   cle.apmaas.agent,oracle.apmaas.repackaged"
   ```

```
set "JAVA_OPTS=%JAVA_OPTS% -javaagent:%JBOSS_HOME%
\apmagent\lib\system\ApmAgentInstrumentation.jar"
```

4. Stop and restart the JBoss Server:

   ```
   % cd <JBOSS_HOME>\bin
   % ./standalone.conf.bat -b 0.0.0.0
   ```

   You can check the entries present in the `<`*`JBOSS_HOME`*`>\startup.log` file to verify that the JBoss Server has started.

5. If you have any managed JBoss application servers, stop and restart them.

Verify APM Java Agent Installation

See Verify APM Java Agent Installation.

6 Install and Configure APM Java Agent on Jetty

Here are the requirements and instructions to install APM Java Agent on Jetty.

Jetty: Requirements and Installation Instructions

Prerequisites for Deploying APM Java Agent on a Jetty Server

- Supported versions:
  - Jetty Server 7, 8, and 9
- JDK version 1.7 or above. If you are using JDK 1.6, either use a JDK 1.6 that supports TLS 1.2 security protocol, or connect to OMC through a Gateway.
- If the JDK version you are running doesn't support TLS 1.2 security protocol, refer to My Oracle Support Doc ID 2703411.1 before proceeding with the APM agent installation.
- You can install one APM Agent in one instance of Jetty.
- **Other considerations:**
  - The machine hosting the Jetty Server should be able to establish an HTTPS connection either directly or indirectly (using a proxy server or an Oracle Management Cloud gateway) to Oracle Management Cloud. For more information about Oracle Management Cloud gateway, see Install a Gateway.
  - The HTTPS connection must use TLS 1.2 security protocol.
  - The install user of APM Java Agent should be the same as the Jetty Server user.
  - The Jetty Server user should have read and write permissions to the directories that host the APM Java Agent, as well as the Jetty Server Home.

Set the `JETTY_HOME` Variable

Set the `JETTY_HOME` variable to point to the Jetty destination directory.

- If you're using a Bash shell:

  ```
  export JETTY_HOME=<Jetty destination directory>
  ```

- If you're using a C shell:

  ```
  setenv JETTY_HOME"<Jetty destination directory>"
  ```

Deploy a Gateway (Optional)

Gateway is not a mandatory component while deploying Oracle Application Performance Monitoring; you can use a gateway in the following scenarios:

- If you have an application server that does not support Transport Layer Security (TLS) protocol 1.2
- If you have older versions of Java Application Servers with JDK less than 1.7 (for example, Oracle WebLogic 10.3.6)

For instructions on how to deploy a gateway, see Install a Gateway.

Set the Gateway Variables (Optional)

Set the values for Gateway host and port.

- If you're using a Bash shell:

  ```
  export GW_HOST=<Gateway Host Name>
  export GW_PORT=<Gateway Port>
  ```

- If you're using a C shell:

  ```
  setenv GW_HOST "<Gateway Host Name>"
  setenv GW_PORT "<Gateway Port>"
  ```

If you are using more than one gateway, use the `-additional-gateways` option with the provisioning script.

Download the APM Java Agent Software for Jetty

1. From the main Oracle Management Cloud menu, navigate to **Administration** and **Agents**.
2. On the Oracle Management Cloud Agents page, click the Action Menu on the top right corner of the page and select **Download Agents**.

   The Agent Software Download page is displayed.
3. From the **Agent Type** dropdown list, select **APM Agent**.
4. Click **APM Java Agent**.
5. Extract the contents of the installer ZIP file.

Install and Provision APM Java Agent on Jetty

- Before you install the APM Java Agent, log in to the machine running the application server as a user who installed the application server.
- The application server user should have Read-Write access to the APM Java Agent directories.

To install and provision the APM Java Agent:

1. Navigate to the directory where you downloaded the APM Java Agent software.
2. Run the provisioning script as per your installation preference:

Installation Preference	Provisioning Script

Basic Installation	`chmod +x ProvisionApmJavaAsAgent.sh`
	`./ProvisionApmJavaAsAgent.sh -d ${JETTY_HOME} -no-wallet`

Silent Installation	`chmod +x ProvisionApmJavaAsAgent.sh`
	`./ProvisionApmJavaAsAgent.sh -d ${JETTY_HOME}-no-prompt`

With Gateway
If you are using more than one gateway, use the `-additional-gateways` option.
	`chmod +x ProvisionApmJavaAsAgent.sh`
	`./ProvisionApmJavaAsAgent.sh -d ${JETTY_HOME} -no-wallet gateway-host {gw_host_value}, -gateway-port {gw_port_value},`
	`-additional-gateways https://<gw_host_1>:<gw_port_1>,https://<gw_host_2>:<gw_port_2>`

In a Proxy Environment	`chmod +x ProvisionApmJavaAsAgent.sh`
	`./ProvisionApmJavaAsAgent.sh -d ${JETTY_HOME} -no-wallet -ph {http_proxy_host} -pp {http_proxy_port} -pt {http_proxy_auth_token}`

- `-d` is the absolute path of the home directory of your Jetty installation. The APM Java Agent will be installed under this directory.
- `-ph {http_proxy_host}` (Optional) — the proxy server's host name.
- `-pp {http_proxy_port}` (Optional) — the proxy server's port.
- `-pt {http_proxy_auth_token}`(Optional) — the authorization token that the agent will use if the proxy server requires authentication. This parameter will be passed literally as the proxy authorization header to the proxy server.

  If you are using HTTP Basic authentication, it is recommended that you omit this parameter. For details on HTTP Basic authentication, see Generating Proxy Token. You can also specify the proxy wallet entry or NTLM credentials token in the following format: `domain/username:password`. For example: `exampledomain/sampleuser:welcome`

The command line displays your tenant name and the value you specified.

3. When prompted, provide the value of the registration key that you've created or downloaded earlier.

   If you are running the provisioning script with the `-no-prompt` option, create a text file containing the value of registration key, and provide the path to the file.

Example: `./ProvisionApmJavaAsAgent.sh -d mydir -no-prompt < regkey.txt` where `regkey.txt` contains a single line with the registration key.

4. Review the values and confirm. To change the values, enter `n` and run the script again with new values.
5. To proceed with the installation, enter `y`.

   The APM Java Agent is installed.

Install APM Java Agent on a Microservice
When running the APM Java Agent provisioning script on an application or microservice using Dropwizard or similar embedded Jetty framework, the installation is very similar to standalone Jetty. First, if your microservice does not have a standard server home (DOMAIN_HOME, CATALINA_HOME, etc.), you can simply choose an empty directory for it. Once it's provisioned, add:

```
-javaagent:<path-to-agent>/apmagent/lib/system/
ApmAgentInstrumentation.jar
```

to the startup command line or script for your application or microservice.

Example:

```
java -javaagent:/u01/apmagent/lib/system/ApmAgentInstrumentation.jar -
jar my-microservice-1.0.jar server config.yml
```

Provision APM Java Agent with Offline Installer
You can install and provision the APM Java Agent from an offline installer that you have received over email.

To install from the emailed ZIP, the provisioning script needs to download the configuration file from Oracle Management Cloud. Specify these additional options while running the provisioning script:

Option	Description
`-tenant-id`	The Oracle Management Cloud tenant name.
`-omc-server-url`	The URL of the Oracle Management Cloud server. Example: `https://omchost:port`

Modify the Startup Script of Your Jetty Server

Modify the startup script of your application server to invoke the Oracle Application Performance Monitoring Cloud Service configuration settings, when you start your application server.

To modify the `java -jar start.jar` script of your Jetty server home:

1. Make a copy of your `java -jar start.jar` file:

   ```
   % cd $JETTY_HOME/bin
   % cp java -jar start.jar java -jar start.jar.orig
   ```

2. Using a text editor, edit the original script to add the `-javaagent` option. To send the Jetty server data to APM Agent, append the following class path, along with `ApmAgentInstrumentation.jar`

   ```
   java -javaagent:/u01/apmagent/lib/system/
   ApmAgentInstrumentation.jar -jar start.jar
   ```

3. Restart the Jetty server.

Verify APM Java Agent Installation

See Verifying APM Java Agent Installation.

7

Install and Configure APM Java Agent on Peoplesoft

Here are the requirements and instructions to install APM Java Agent on Peoplesoft.

Peoplesoft: Requirements and Installation Instructions

Prerequisites

- PeopleSoft running on PeopleTools 8.55 with WebLogic container for PeopleSoft UI — PeopleSoft Internet Architecture (PIA)
- If the JDK version you are running doesn't support TLS 1.2 security protocol, refer to My Oracle Support Doc ID 2703411.1 before proceeding with the APM agent installation.
- The HTTPS connection must use TLS 1.2 security protocol.
- Download the APM Java Agent software referring to the install instructions for the APM Java Agent for WebLogic Server. See Install and Configure APM Java Agent On Oracle Weblogic Server.
- Install the APM Java Agent software.

After installing the APM Java Agent software, perform these steps to provision the APM Java Agent on PeopleSoft:

1. Navigate to the PIA Domain Administration Menu:
   1. From the shell prompt, start the psadmin utility with the following command:
   2. Select **4** for Web (PIA) Server.
   3. Select **1** to administer a domain.
   4. Select the PIA domain where to install the APM Agent.
2. Edit the configuration files:
   1. View the list of configuration files to edit, by executing command **5**.
   2. Select **1** to edit the `setEnv` file. The `setEnv` file will open in a Vi editor.
   3. Search for the corresponding `JAVA_OPTIONS_<Operating System>` variable and add the following option:

      ```
      JAVA_OPTIONS_<Operating System>="${JAVA_OPTIONS_<Operating
      System} -javaagent:${DOMAIN_HOME}/apmagent/lib/system/
      ApmAgentInstrumentation.jar" where <Operating System> is the
      operating system that PeopleSoft PIA runs on.
      ```

   4. Save the changes made to the `setEnv` file and exit the Vi editor, to see the list of configuration files.

   e. Type **q** to exit this menu, and to see the PIA Administration menu.

3. Reboot the PIA domain:
   a. Navigate to the PIA Domain Administration menu. Refer to step 1.
   b. Select **2** to shut down the domain.
   c. Select **3** to get the status of the domain to check if it is still running.
   d. Select **1** to boot this domain to complete the restart of the PIA domain.

Peoplesoft: Custom Instrumentation Set Up for Tuxedo

You can set up custom instrumentation to add monitoring capabilities to Peoplesoft.

To set up custom instrumentation for Tuxedo in Peoplesoft, do the following:

- Create a file named *custom-pointcuts.properties* with the following entries:

  ```
  #Pointcut for JOLT
  pointcut-jolt.class = bea.jolt.JoltRemoteService
  pointcut-jolt.method = call
  pointcut-jolt.paramTypes = bea.jolt.Transaction
  pointcut-jolt.paramsTypes = bea.jolt.Transaction
  pointcut-jolt.value.exit.jolt-service.path = this.getName()
  pointcut-jolt.operationName = Tuxedo_{value.jolt-service}
  pointcut-jolt.tier = external
  ```

  Save the *custom-pointcuts.properties* file under the configuration directory for the specific WebLogic Server in the apmagent directory. For example, configuration directory: `<psft_home>/psadm2/psft/pt/8.56/webserv/peoplesoft/apmagent/config/PIA/`

- Update file: `AgentStartup.properties` located under `<APM_agent_install_dir>/config` directory and add entry:

  ```
  oracle.apmaas.agent.probe.custom.enableMethodInvocation = true
  ```

- Restart the application server for the configuration changes to take effect.

After the pointcut is enabled, the Tuxedo wait time (the time it takes the Tuxedo servers to request and get the data from the database) will be marked as external. It will be visible in the server request list and detail pages, and also in the topology views. In addition, the name of the requested `Tuxedo` service will be available in the aggregated call stack view.

For more details, see Set Up Custom Instrumentation from *Using Oracle Application Performance Monitoring*.

8

Install and Configure APM Java Agent on Oracle E-Business Suite

You can monitor the performance of Oracle E-Business Suite applications using Oracle Application Performance Monitoring.

Oracle E-Business Suite: Requirements and Installation Instructions

You can install Oracle APM Java Agent to monitor Oracle E-Business Suite (EBS) applications.

Prerequisites for Deploying APM Java Agent on Oracle E-Business Suite

- The machine hosting Oracle E-Business Suite should be able to establish an HTTPS connection either directly or indirectly (using a proxy server or an Oracle Management Cloud gateway) to Oracle Management Cloud. For more information about Oracle Management Cloud gateway, see Install a Gateway.
- The HTTPS connection must use TLS 1.2 security protocol.
- If the JDK version you are running doesn't support TLS 1.2 security protocol, refer to My Oracle Support Doc ID 2703411.1 before proceeding with the APM agent installation.
- The install user of APM Java Agent should be the same as Oracle E-Business Suite install user.
- The Oracle E-Business Suite user should have read and write permissions to the directories that host the APM Java Agent.

Download the APM Java Agent Software for Oracle E-Business Suite

1. From the Oracle Management Cloud menu, navigate to **Administration** and **Agents**.
2. On the Oracle Management Cloud Agents page, click the Action Menu on the top right corner of the page and select **Download Agents**.

   The Agent Software Download page is displayed.
3. From the **Agent Type** dropdown list, select **APM Agent**.
4. Click **APM Java Agent**.
5. Extract the contents of the installer ZIP file.

Install and Provision APM Java Agent on Oracle E-Business Suite

- Before you install the APM Java Agent, log in to the machine running the application server as a user who installed the application server.

- The application server user should have Read-Write access to the APM Java Agent directories.

To install and provision the APM Java Agent:

1. Navigate to the directory where you downloaded the APM Java Agent software.
2. Run the provisioning script:

   ```
   chmod +x ProvisionAPMJavaAsAgent.sh
   $ ./ProvisionApmJavaAsAgent.sh -d <PATH_TO_fs_ne> -no-wallet
   ```

   - `-d` is the absolute path of the home directory of the WebLogic Server domain. The APM Java Agent will be installed under this directory.
   - `<PATH_TO_fs_ne>` is the full path for the non edition directory. For example: `/u01/ebs122/fs_ne/`. Alternatively, any other non edition location can be used (i.e., not under fs1 or fs2).
   - Note the installation shown here is for an environment without a gateway or a proxy.
3. When prompted, provide the value of the registration key that you've created or downloaded earlier.
4. Review the values and confirm. To change the values, enter `n` and run the script again with new values.
5. To proceed with the installation, enter `y`.

   The APM Java Agent is installed.

Enable the APM Java Agent in the WebLogic Managed Servers

To enable the access to Oracle E-Business Suite from Oracle Application Performance Monitoring, JVM configuration changes must be made to the WebLogic Managed Servers. Perform the following steps on each of the managed servers of the **oacore**, **oafm**, **forms** and **forms-c4ws** services:

1. Log in to the WebLogic Server Administration Console as a user with admin security role. By default, it’s the user `weblogic`.
2. Click **Servers**. The WebLogic Administration Server and Managed Servers summary page is displayed.
3. Choose the managed server for which to change the configuration - **oacore**, **oafm**, **forms** or **forms-c4ws**. If you need to monitor the Java applet traffic in Oracle E-Business Suite Forms Applications then also include the forms server and then refer to Deploy the APM Java Agent with Oracle Forms Monitoring.
4. A page containing various tabs for the settings of the managed server appears.

   Click **Server Start** tab.

5. In the **Change Center**, click **Lock and Edit**. Update the **Arguments** field with the parameters required for Oracle Application Performance Monitoring. Ensure that the existing arguments are not altered.

   ```
   -javaagent:<PATH_TO_fs_ne>/apmagent/lib/system/
   ApmAgentInstrumentation.jar
   ```

6. Click **Save**.
7. In the **Change Center**, click **Activate Changes** to activate the changes.
8. Repeat all the above steps for each of the managed servers of the **oacore**, **oafm**, **forms** and **forms-c4ws** services, till each managed server has the JVM arguments updated.
9. Restart the managed servers using the script available at `$ADMIN_SCRIPTS_HOME` in the Oracle E-Business Suite environment for stopping and starting the managed servers.

Once you have successfully installed an APM Agent, you can enable End User Monitoring in an Oracle E-Business Suite environment by configuring browser agents with **Reference** injection type for APM Agents related to **oacore** managed servers. See Enable and Configure End User Monitoring.

Configure User Name Reporting for Oracle E-Business Suite

To configure user name reporting for Oracle E-Business Suite, do the following:

1. Create a file in the `$OA_HTML` directory on the EBS server. For example `omc_username.htm`.
2. Insert one of the javascript tags described under get username from EBS Username on Configure User Name Reporting.

   The exact tag to be used depends on details of your EBS deployment.

   For example, you can insert the following javascript snippet:

   ```
   <head>
   ...
   ...
   ...
   </head>
   <body>
   ...
   ...
   ...
   <script type='text/javascript' charset='UTF-8'>
     var namefromCookie = apmeum.util.getCookie('EBSUSERNAME');
     if (namefromCookie != null && namefromCookie.length > 0) {
       apmeum.username = namefromCookie;
     } else {
       var spanList1 = document.getElementsByClassName("x1f");
       var spanList2 = document.getElementsByClassName("x2u");
   ```

```
    if (spanList1 != null && spanList1.length > 0) {
      var loginName = spanList1[0].innerHTML;
      apmeum.username = loginName.replace('welcome ','');
    } else if (spanList2 != null && spanList2.length > 0) {
      var loginName = spanList2[0].innerHTML;
      apmeum.username = loginName;
    }
  }
</script>
</body>
```

The above javascript snippet captures the `EBSUSERNAME` EBS user name value and stores it into a cookie in case there are pages where the EBS user name variable isn't present, but you still require the value of the EBS user name.

3. Log into EBS as the SYSADMIN user and select the **Functional Administrator Responsibility** option.
4. Navigate to Personalization tab and enter the path `/oracle/apps/fnd/sso/login/webui` and click **Go** to retrieve the document.
5. Select the **Personalize Page** icon for the MainLoginPG document.
6. Accept the defaults on the **Choose Personalization Context** screen and click **Apply**. Note that the scope is now `/oracle/apps/fnd/sso/login/webui/MainLoginPG`.
7. Select the **Complete View** option under **Personalization Structure**, and click on the **Create Item** icon for the **Page Layout** row.
8. On the **Create Item** form, do the following:
   - Set **Item Style** to **URL include**.
   - Enter a name for the **ID**, for Example `OMC_USERNAME`.
   - Set Source **URI** to `/OA_HTML/omc_username.htm`. (This is the file created in Step 1).
   - Click **Apply** to save changes.

   On the **Personalization** page you will see the "include url" item added using the steps above.
9. Verify in a new browser window that the name is captured and showing up as a cookie, as well as in the information send via a POST command to the OMC collector.

Deploy the APM Java Agent with Oracle Forms Monitoring

You can deploy APM Java Agent with Oracle Forms Monitoring in Oracle E-Business Suite.

Prerequisites

- Oracle Forms server needs to be configured in servlet mode: A Java servlet, called the Forms Listener servlet, manages communication between the Forms Java client and the OracleAS Forms services.

  To check if Oracle Forms server is running in servlet mode, do the following:

  - Set the environment variables for Oracle E-Business Suite using the script `EBSapps.env`.
    For example: `source /u01/install/APPS/EBSapps.env run`
  - Check if servlet mode is configured by inspecting the value of `connectMode` parameter from the Forms configuration file.

    ```
    $ grep connectMode $FORMS_WEB_CONFIG_FILE
    connectMode=servlet
    ```

    For more information about checking if Oracle Forms server is running in servlet or socket mode, see Checking Socket and Servlet Mode.
- Ensure that the Forms Server is sending the `Form Name` inside the Forms protocol.

  In order to do that, you need to configure the parameter `FORMS_RUEI_SEND_FORM_NAME=TRUE` in the `default.env` file or the `$APPL_TOP/APPS<S_CONTEXTNAME>.env` file.
- Ensure that Forms is patched by checking if parameter: `FORMS_RUEI_SEND_FORM_NAME` is available.

  If the following command returns `FORMS_RUEI_SEND_FORM_NAME` then the Forms environment is patched.

  ```
  $ strings $ORACLE_HOME/lib/libiffw.so | grep RUEI
  FORMS_RUEI_SEND_FORM_NAME
  ```
- Check if Forms server is running and the environment is set correctly by running the following command:

  ```
  $ strings /proc/$(pgrep -x frmweb)/environ | grep RUEI
  FORMS_RUEI_SEND_FORM_NAME=TRUE
  ```

  If the command returns: `strings: '/proc//environ': No such file` then the `frmweb` process is not running and you need to start it to make sure that it is running. You may need to have a LIVE applet session running in order for this process to exist.

Deploy the APM Java Agent with Oracle Forms Monitoring

1. Follow the instructions for Download the APM Java Agent Software for Oracle E-Business Suite and Install and Provision APM Java Agent on Oracle E-Business Suite.
2. Follow the instructions for Enable the APM Java Agent in the WebLogic Managed Server and deploy the APM Java Agent to the managed server. The Forms managed server is usually called **forms_server1**.
3. Start the managed server: **forms_server1**.

4. Verify that the last line in `<domain-home>/apmagent/logs/forms_server1/AgentStartup.log` looks similar to the following:

   ```
   0xd<2019-03-20T11:37:18.800-0700> INFO <STARTUP> Agent startup
   successfully completed - the agent is now operational and
   monitoring traffic
   ```

APM Java Agent supports Oracle Forms monitoring only with Oracle E-Business Suite release 12. Older releases are not supported.

9

Install and Configure APM .NET Agent

Topics:

Agent Requirements and Installation Instructions

Prerequisites

- Windows Server 2008 and above
- .Net Framework 3.5 and above

> **Note:**
>
> If your APM is installed on .Net Framework 3.5 Classic, then, APM does not monitor the static file request page. The .Net Framework 3.5 Integrated and other higher versions of the .Net framework supports static file request monitoring by APM.

- Java Application Servers with JDK version 1.7 or above. If you are using JDK 1.6, either use a JDK 1.6 that supports TLS 1.2 security protocol, or connect to OMC through a Gateway.
- If the JDK version you are running doesn't support TLS 1.2 security protocol, refer to My Oracle Support Doc ID 2703411.1 before proceeding with the APM agent installation.
- For monitoring of .NET applications on Windows, only one APM product should be configured on a system at a time.
- **Other considerations:**
  - The machine hosting the IIS Server should be able to establish an HTTPS connection either directly or indirectly (using a proxy server or an Oracle Management Cloud gateway) to Oracle Management Cloud. For more information about Oracle Management Cloud gateway, see Install a Gateway.
  - The HTTPS connection must use TLS 1.2 security protocol.
  - The users that run applications in IIS should have *Read* and *Write* permissions to the .Net agent log directories. By default, the log directory is `C:\ProgramData`.
  - The user should have Administrator Access to the machine where APM .Net Agent will be deployed.

- The user should have the ability to restart IIS.

Set the Registration Key

After creating or downloading a registration key, set the `REG_KEY` variable to the registration key.

```
set REG_KEY=<Registration Key>
```

Deploy the Gateway (Optional)

A gateway is not a mandatory component while deploying Oracle Application Performance Monitoring. Use the gateway in the following scenarios:

- If you have an application server that does not support Transport Layer Security (TLS) protocol 1.2.
- If you have older versions of .NET IIS servers and Java Application Servers with JDK less than 1.7 (for example, Oracle WebLogic 10.3.6)

For instructions on how to deploy the gateway, see Instal a Gateway in *Installing and Managing Oracle Management Cloud Agents*.

Set the Gateway Variables (Optional)

Set the values for Gateway host and port.

```
set GW_HOST "<Gateway Host Name>"
set GW_PORT "<Gateway Port>"
```

Download the APM .NET Agent Software

1. From the main Oracle Management Cloud menu, navigate to **Administration** and **Agents**.
2. On the Oracle Management Cloud Agents page, click the Action Menu on the top right corner of the page and select **Download Agents**.

   The Agent Software Download page is displayed.
3. From the **Agent Type** dropdown list, select **APM Agent**.
4. Click **APM .Net Agent**.
5. Extract the contents of the installer ZIP file.
6. Click **Done** after the download is complete.
7. Create a registration key that will be used during the time of installing a new agent. Oracle Application Performance Monitoring Cloud Service verifies this key before accepting any data sent by APM .Net Agent deployed on your on-premises hosts. For more information about creating a registration key, see Manage Registration Keys in *Installing and Managing Oracle Management Cloud Agents*.
8. Enable the performance counters of the IIS application you wish to monitor. For this, the application pool user must be a member of the 'Performance Monitor Users' group. Refer to the Microsoft IIS documentation for detailed instructions.

Install the APM .NET Agent Software

To install the APM .Net Agent on your IIS server:

1. Make a backup of the following system configuration files:
   - `C:\Windows\System32\inetsrv\config\applicationHost.config`
   - `C:\Windows\Microsoft.NET\Framework\v4.0.30319\Config\web.config`
   - `C:\Windows\Microsoft.NET\Framework\v2.0.50727\CONFIG\web.config`
   - `C:\Windows\Microsoft.NET\Framework64\v4.0.30319\Config\web.config`
   - `C:\Windows\Microsoft.NET\Framework64\v2.0.50727\CONFIG\web.config`
2. Stop the IIS server.

   ```
   iisreset /stop
   ```
3. Execute the APM .Net Agent installer.
4. Ensure that the `AgentConfig.info` file is in the same directory as the `ApmAgent.msi` file.
5. Run the `ApmAgent.msi` executable. The APM .Net Agent installation wizard guides you through the installation process.
6. The Installation wizard performs a check of these pre-requisites before proceeding with the installation:
   a. User without administration privileges
   b. If IIS is installed
   c. If IIS is installed without ASP .Net
   d. If the user does not have the permission to access this path — `"C:\Windows\System32\inetsrv\config\applicationHost.config"`

   A warning is displayed if any of the pre-requisites are not met.
   - Choose **No** to exit the installation.
   - Choose **Yes** to proceed anyway.

   However, if you proceed without all the pre-requisites in place, the installation of the APM .Net Agent will not be successful.
7. Specify the installation directory.
8. Optionally, select **Configure Proxy** and specify the proxy information. The password that you specify here is encrypted to enhance security.
9. Optionally, select **Configure Gateways** and specify the Gateway information.
10. In the **Set Registration Key** field, provide the Registration Key. You can get this from the **Registration Keys** tab in the Oracle Management Cloud Agents page.
11. In the **Set Host Name** field, review the default host name, and modify if required.
12. Select the applications to be monitored in the **Monitored Application Configuration** screen. You can choose to monitor all applications on the IIS server, or manually select specific applications from the list.

    You can change this list of applications to be monitored anytime after the installation. See Enable Monitoring of Specific Applications.
13. Click **Install**.

14. For Silent Installation, run this command:

    ```
    msiexec /i <msi file name> /quiet /log <log name to redirect
    the installation output> REGISTRATION_KEY=<reg key value>
    AGENTCONFIGEXISTS=1
    ```

    > **Note:**
    >
    > For a silent mode install with proxy, run this command instead:
    >
    > ```
    > msiexec /quiet -i <msi file name> SERVER_URL=<the server
    > url> TENANT_ID=<the tenant ID> REGISTRATION_KEY=<reg
    > key value> PROXY_ENABLE=1 PROXY_HOST=<the proxy url>
    > PROXY_PORT=<the proxy port> PROXY_USER=<proxy user name>
    > PROXY_PASSWORD=<the proxy password>
    > ```

15. Start your IIS server.

    ```
    iisreset /start
    ```

> **Note:**
>
> APM .NET Agent uses .NET Profile API like most of the IIS monitoring tools. This might lead to potential conflicts when APM .NET Agent is installed alongside another IIS monitoring tool. As a result, both APM .NET Agent and another tool may not be able to detect or correctly monitor IIS application that needs to be monitored.

Enable Monitoring of Specific Applications

You can configure APM .Net Agent to monitor specific applications installed on your IIS server. You can choose the applications to be monitored during the installation. You can also choose to add a new application to be monitored, or stop monitoring some applications after the installation.

To manually configure applications to be monitored by APM .Net Agent:

1. Disable monitoring globally for the APM .Net Agent.
    a. In the installation directory, check the global `AgentConfig.ini` file. By default, the installation directory is `C:\Program Files\Oracle APM .NET Agent`.
    b. Disable global monitoring for all applications with this setting:

        ```
        oracle.apmaas.agent.disable=true
        ```

2. Enable monitoring for a specific application.
    a. Navigate to the Agent’s instance directory: `C:\ProgramData\Oracle\ApmAgent\config\<webSite><webApplication>`.

b. Edit the `AgentConfig.ini` file and enable monitoring with this setting:

```
oracle.apmaas.agent.disable=false
```

Repeat this for every application to be monitored.

Verify the APM .NET Agent Installation

1. Use the Oracle Application Performance Monitoring Web Console
2. Examine the APM .Net Agent logs located at `C:\ProgramData\Oracle\ApmAgent\logs\<site>\`.

   Note that the log files are not initially created, even if the installation of APM was successful. The log files get created and populated only when the application you are monitoring is running, and shows some activity which results in data depicted on the Oracle Application Performance Monitoring Web UI.

   - If the agent logs are empty without any data, check the Windows **Event Viewer** for any possible problems with the agent installation. (See **Custom Views** and select **Administrative Events**.)
   - If you see the following message

     The underlying connection was closed: Could not establish trust relationship for the SSL/TLS secure channel

     in the agent logs, it means that the Oracle Management Cloud certificate was not provisioned since the certificate was not available. Add the root certificate `emcs.cer` to the trusted root certification authorities on your local machine. See https://technet.microsoft.com/en-in/library/cc754841.aspx#BKMK_addlocal.

Configure APM .NET Agents

You can configure deployment parameters of your APM .Net Agent to control the monitoring of applications.

- APM .Net Agent Installation Directories
- Configuration Directory
- Logs Directory
- Important Considerations
- Configuration Settings
- Proxy and Gateway Settings
- Disable Browser Agent Monitoring by APM .Net Agent

APM .Net Agent Installation Directories

The APM .Net Agent installation directory has an installation directory, configuration directory and logs directory.

The agent installation directory is specified by the environment variable ORACLE_APM_AGENT_HOME, which gets automatically set by the installer. Each agent instance contains configuration directory and logs directory.

Examples:

Installation Directory: `c:\Program Files\Oracle APM .Net Agent`

Configuration directory:`c:\ProgramData\Oracle\ApmAgent\config\<appName>`

Logs directory: `c:\ProgramData\Oracle\ApmAgent\logs\<appName>`

Configuration Directory

By default, the agent configuration directory for a monitored application is `c:\ProgramData\Oracle\ApmAgent\config\<appName>`.

You can customize the configuration of an APM .Net Agent instance by adding an `AgentConfig.ini` file to the configuration directory. By customizing the configuration you can disable or turn log traces for a specific agent.

After you have made the configuration changes, only the agent instance using this configuration directory will detect the configuration change and restart. If several agent instances are monitoring the same replicated application, all agent instances will detect the configuration change and restart.

Logs Directory

The APM .Net Agent logs directory contains three log files: `AgentStartup.log`, `AgentStatus.log` and `Agent.log`.

- `AgentStartup.log` contains the agent logs related to startup. You can verify if the agent started up correctly from this file.
- `Agent.log` contains all agent logs, including the logs found in `AgentStartup`.

The change the level of logging in these files, see Configuration Settings.

Important Considerations

Some important considerations before configuring the parameters:

1. Any change to the installed configuration files (`AgentConfig.ini` and `OMC.ini`), applies to all agent instances running on the machine.
2. An agent automatically detects any change to any of its configuration files. When this happens an agent restarts, making any new setting immediately active. If a change is made to the installation directory configuration files (`AgentConfig.ini` or `OMC.ini`), all agents on the machine restart reading the changed configuration files as they start up. If a specific `AgentConfig.ini` configuration file gets changed (for example, from an agent instance configuration directory), only the specific agent restarts.

Configuration Settings

Property	Description
`oracle.apmaas.agent.disable`	Defaults to `False` when not specified. If set to `True`, the agent is disabled (agent instance is shutdown).

Property	Description
`oracle.apmaas.agent.trace.bootstrap.limitoracle.apmaas.agent.trace.bootstrap.rotation`	`AgentBootstrap` log file settings — • Rotation default is 2 • Limit default is 20 Mb • Limit indicated in Mbs (Example: 10 for 10 Mb)
`oracle.apmaas.agent.trace.startup.limit` `oracle.apmaas.agent.trace.startup.rotation`	`AgentStartup` log file settings — • Rotation default is 2 • Limit default is 20 Mb • Limit indicated in Mbs (Example: 10 for 10 Mb)
`oracle.apmaas.agent.trace.status.limit` `oracle.apmaas.agent.trace.status.rotation`	`AgentStatus` log file settings — • Rotation default is 2 • Limit default is 20 Mb • Limit indicated in Mbs (Example: 10 for 10 Mb)
`oracle.apmaas.agent.trace.limit` `oracle.apmaas.agent.trace.rotation`	`Agent` log file settings — • Rotation default is 2 • Limit default is 20 Mb • Limit indicated in Mbs (Example: 10 for 10 Mb)
`oracle.apmaas.agent.trace.<area> = <level>`	`area: Instrumentation, Startup, Http, MetricsProcessor, Sampling, Handler, Transport, Config, InternalMetrics, SqlServer, Utility, Injection level: verbose, all, off`

`oracle.apmaas.agent.trace = <level>`	`All trace areas`
	`level: verbose, all, off`

`oracle.apmaas.agent.trace.flow = verbose`	When the trace area flow is enabled with the level `verbose`, detailed traces of the flow processor are logged in a file named `AgentFlowTrace.log`. See section below for details.
`oracle.apmaas.agent.trace.overhead = verbose | all`	Enable this to get a dedicated log file named `AgentOverheadStatus.log` created. The file contains metrics on probe, transport (`verbose`) and optionally JSON, core runtime (all) internal processing overhead metrics, and some memory metrics. These metrics can be used to get the time that the agent adds (overhead) to the monitored application, in microseconds. Note that fetching these metrics also adds overhead, so the real overhead is few microseconds less than the logged overhead.

Property	Description
`oracle.apmaas.agent.trace.overhead.format = csv | json`	Set this property value to `csv` to get a Comma Separated Values output format instead of a user readable text format (this applies to the `AgentOverheadStatus.log` file). Set this value to `json` to create separate JSON files. If the `json` format is specified, the file `AgentOverheadStatus.log` is not created.
`oracle.apmaas.agent.trace.observations = verbose`	Set this property to log all observations sent by the agent in a local file, within the agent log directory. By default, the file name is `Observations.log`. Use one of properties below instead of this one to specify a different file name.

`oracle.apmaas.agent.testServerUrl = file | url`
OR
`oracle.apmaas.agent.trace.observations.locationUrl = file | url`
	Set any of these properties to a file name to log observations within the specified file name in the agent log directory. If a URL is specified (typically the URL `AgentCLI` listens on), observations will be sent to the specified URL. If the agent is setup to communicate with cloud services, observations will be sent both to cloud services and to the specified URL. To disable communication with cloud services and only have the agent communicate with the specific URL, set the property `oracle.apmaas.agent.test.skipCloudStartup=true`
	Note that if you log observations within a file, you can also use the following setting:
	`oracle.apmaas.agent.trace.observations.maxSizeMB (same as limit, defined for Java cross platform compatibility) oracle.apmaas.agent.trace.observations.limit (same semantic as limit described above) oracle.apmaas.agent.trace.observations.rotation (same semantic as rotation described above) oracle.apmaas.agent.trace.observations.prettyPrint (set it to true, to pretty print logged observations)`

`oracle.apmaas.agent.identity.concurrent.max`	Maximum number of agents allowed to run concurrently (should be equal to `max_worker_proecess` defined in `ApplicationPool`. The default value is 1.

Property	Description
`oracle.apmaas.agent.lock.identity.waitInMs`	When all agent identities are being used, this property defines how long (in ms) the .Net Agent will wait before trying to retrieve agent identity again. The default value is 10 seconds.
`oracle.apmaas.agent.lock.identity.waitTimesToWriteLog`	When all agent identities are being used and the .Net Agent is trying to retrieve agent identity again, this property defines after how many times of trying, the .Net Agent will write a log to inform user that it is still trying. The first trying will always be logged. The default value is 120.

Proxy and Gateway Settings

If you are installing and provisioning the APM .Net Agent in environments that require the use of proxy servers, use these additional options. Contact your network administrator for these values.

Proxy Parameters

Option	Description
`oracle.apmaas.agent.proxyHost`	The proxy server's host name. This is an optional parameter.
`oracle.apmaas.agent.proxyPort`	The proxy server's port. This is an optional parameter.
`oracle.apmaas.agent.proxyAuthUser`	This is the user name the agent will use if the proxy server requires authentication.
`oracle.apmaas.agent.proxyAuthPassword`	This is the password required if the proxy server requires authentication.
`oracle.apmaas.agent.proxyAuthDomain`	This is the name of the domain if the proxy server requires authentication.

If you are installing and provisioning the APM .Net Agent in environments that require the use of Gateway, use these additional options. Contact your network administrator for these values.

Gateway Parameters

Option	Description
`oracle.apmaas.agent.uploadRoot`	The APM agent's root directory.
`oracle.apmaas.agent.collectorRoot`	The APM agent's collector root.

Disable Browser Agent Monitoring by APM .Net Agent

In the APM .Net Agent's configuration file, make the following settings:

- To disable monitoring: `oracle.apmaas.agent.enableBrowserAgent=false`
- To disable specific URLs: `oracle.apmaas.agent.jsinjection.disableRegex=<regex>`

- Add `oracle.apmaas.agent.browser.observations`

10 Install and Configure APM Node.js Agent

Topics:

Agent Requirements and Installation Instructions

Prerequisites

- Any unzip utility.
- The machine where you install the APM Node.js Agent should be able to establish an HTTPS connection either directly or indirectly (using a proxy server or an Oracle Management Cloud gateway) to Oracle Management Cloud. For more information about Oracle Management Cloud gateway, see Install a Gateway.
- The HTTPS connection must use TLS 1.2 security protocol.
- The install user should have read-write access to the directories that host the APM Node.js Agent (<*NODE_HOME*/`lib/node_modules`>) and to the *`STAGE_DIR`* (directory where the APM Node.js Agent installer ZIP file is extracted).

Set the `NODE_PATH` Variable

Set the `NODE_PATH` variable to point to `node_modules`:

- On Linux:

  ```
  <Node Installation Directory>/lib/node_modules
  ```

- On Windows:

  ```
  <USER_HOME>\AppData\Roaming\npm\node_modules
  ```

Set the Proxy Variables (Optional)

If you are trying to deploy the APM Node.js Agent over a proxy server, then you need to set the proxy variables: `http_proxy` and `https_proxy` on the host where you are deploying the agents.

If you are using a Bash shell:

- `export http_proxy=http://www-hostname.abc.com:<port>/`
- `export https_proxy=http://www-hostname.example.com:<port>/`

If you're using a C shell:

```
setenv http_proxy “proxy”
```

Deploy the Gateway (Optional)

A gateway is not a mandatory component while deploying Oracle Application Performance Monitoring. Use the gateway in the following scenarios:

- If you have an application server that does not support Transport Layer Security (TLS) protocol 1.2
- If you have older versions of .NET IIS servers and Java Application Servers with JDK less than 1.7 (for example, Oracle WebLogic 10.3.6)

For instructions on how to deploy the gateway, see Installing a Gateway in *Installing and Managing Oracle Management Cloud Agents*.

Set the Gateway Variables (Optional)

Set the values for Gateway host and port.

- If you're using a Bash shell:

  ```
  export GW_HOST=<Gateway Host Name>
  export GW_PORT=<Gateway Port>
  ```

- If you're using a C shell:

  ```
  setenv GW_HOST "<Gateway Host Name>"
  setenv GW_PORT "<Gateway Port>"
  ```

Download the APM Node.js Agent Software

1. On the Oracle Management Cloud home page, click the Oracle Management Cloud Navigation icon on the top-left corner to view the Management Cloud navigation pane.
2. Select **Administration** and **Agents**.
3. On the Oracle Management Cloud Agents page, click the Action Menu on the top right corner of the page and select **Download Agents**.

   The Agent Software Download page is displayed.
4. From the **Agent Type** drop-down list, select **APM Agent**.
5. Click **APM Node.js Agent**.

   This downloads the agent software ZIP file to the selected location.
6. Create a registration key that will be used during the time of installing a new agent. Oracle Application Performance Monitoring Cloud Service verifies this key before accepting any data sent by APM Node.js Agent deployed on your on-premises hosts. For more information about creating a registration key, see Manage Registration Keys in *Installing and Managing Oracle Management Cloud Agents*.
7. Extract the contents of the installer ZIP file to an empty directory (for example, `STAGE_DIR`).

Install and Provision APM Node.js Agent

1. Navigate to the directory where you downloaded the APM Node.js Agent software (STAGE_DIR).
2. Execute the APM Node.js Agent installer.

OS	Example

On Linux:	`chmod +x ProvisionApmNodeAgent.sh`
	`./ProvisionApmNodeAgent.sh`

On Windows:	`ProvisionApmNodeAgent.cmd`

3. The APM Node.js Agent installer prompts for the registration key. Provide the registration key and press Enter. The APM Node.js Agent gets installed.

   For a list of options supported by the APM Node.js Agent provisioning script use: Example:

OS	To get help
On Linux:	`./ProvisionApmNodeAgent.sh -help`
On Windows:	`ProvisionApmNodeAgent.cmd /?`

4. The Provisioning script will compute a default `hostname` and use that as one of the ID attributes. To override the default, run the provisioning script with the `ORACLE_HOSTNAME` argument.

   Example:

OS	To override the default hostname
On Linux:	`./ProvisionApmNodeAgent.sh -h <hostname>`
On Windows:	`ProvisionApmNodeAgent.cmd /h <hostname>`

`hostname` will be one of the identity parameters of APM Node.js Agent which will be visible on the Oracle Management Cloud Application Performance Monitoring UI.

5. If you are installing and provisioning the APM Node.js Agent in environments that require the use of proxy servers, use these additional options. Contact your network administrator for these values.

```
./ProvisionApmNodeAgent.sh [-ph <proxy host>][-pp <proxy port>][-pt
<proxy token>]
```

Proxy Option	Description
`-ph`	The proxy server host name. This is an optional parameter. **Example:** `-ph myproxyhost.example.com`
`-pp`	The proxy server port number. This is an optional parameter. **Example:** `-pp 80`
`-pt`	This is the authorization token that the agent will use if the proxy server requires authentication. This is an optional parameter that will be passed literally as the proxy authorization header to the proxy server.

6. If you are installing and provisioning the APM Node.js Agent with a gateway, use these additional values.

```
./ProvisionApmNodeAgent.sh [-gateway-host <gateway host>] [ -
gateway-port <gateway port>] -additional-gateways <additional
gateways>
```

Gateway Option	Description
`-gateway-host`	The gateway host through which the APM Node.js Agent communicates with the Oracle Management Cloud server. **Example:** `-gateway-host mygateway.example.com`
`-gateway-port`	The gateway agent port number. This is an optional parameter. **Example:** `-gateway-port 4459`
`-additional-gateways`	Comma separated list of gateway URLs. **Example:** `-additional-gateways https://mygateway2.example.com:1896,https://mygateway3.example.com:1897`

Also see Additional Gateway Options on Microsoft Windows

7. If you are installing APM Node.js Agent using a silent installation (unattended) option, use the following command:.

```
/ProvisionApmNodeAgent.sh -regkey-file <regkey file> [-tenant-id
<tenant name>] -no-prompt
```

Silent Installation Option	Description
`-regkey-file <regkey file>`	The registration key information where `<regkey file>` is the filename of the registration key file.
`-tenant-id <tenant name>`	The tenant id information where `<tenant name>` is the tenant name.
`-no-prompt`	No user interaction is required.

Provision APM Node.js Agent with Offline Installer

You can install and provision the APM Node.js Agent from an offline installer that you have received over email.

To install from the emailed ZIP, the provisioning script needs to download the configuration file from Oracle Management Cloud. The `DownloadApmNodeAgentConfiguration.sh` script downloads the latest configuration from Oracle Management Cloud and generates the APM Node.js Agent configuration file `oracle-apm-config.json`.

Specify these additional options while running the provisioning script:

Option	Description
`-tenant-id`	The Oracle Management Cloud tenant name.
`-omc-server-url`	The URL of the Oracle Management Cloud server. Example: `https://omchost:port`

Additional Gateway Options on Microsoft Windows

This version of APM Node.js Agent does not support Gateway options over Microsoft Windows.

If you are installing APM Node.js Agent over Microsoft Windows, follow these steps to configure additional gateways:

1. Ensure that the APM Node.js Agent is provisioned with Oracle Management Cloud/primary gateway successfully using the above steps.
2. Verify the installation. See steps below.
3. Manually copy the additional gateway/s certificates into the folder which has the extracted agent zip file.
4. To provision the additional gateway/s certificates, execute the below command:

```
 cp "${CERT_FILE}" "<CERT_FILE_NAME>".der
oracle-apm keytool der2pem "<CERT_FILE_NAME>".der "<CERT_FILE_NAME>"
oracle-apm update data "<CERT_FILE_NAME>"
```

5. Navigate to the Node.js application home and execute the below commands to get the value of the current `uploadRoot`:

```
mkdir -p oracle-apm/config
oracle-apm config init
oracle-apm config get uploadRoot
```

   The value of the current `uploadRoot` is displayed as below. The new `uploadRoot` will be displayed with the existing value, and the new URLs, separated by commas.

```
oracle-apm config set uploadRoot <New value for uploadRoot>
```

6. To confirm that the new value has been updated, run

```
oracle-apm config get uploadRoot
```

Verify the Installation:

1. Ensure that the `oracle-apm` directory is created in the `node_modules` directory of your Node installation as below.

   On Linux: `$NODE_HOME/lib/node_modules`

   On Windows: `<USER_HOME>\AppData\Roaming\npm\node_modules`

2. In the `node_modules` directory within `oracle-apm`, a folder called `data` is created with the following files:

   - `oracle-apm-config.json`
   - `emaas.cer`
   - Gateway certificates, if the agent is provisioned with the gateway.

     Example: `trustCertGateway.cer`

Troubleshooting
If you see this error — `npm ERR! code 1 - error`, it is an indication that `oracle-apm` agent was not installed previously. You can proceed with the installation.

Verify APM Node.js Agent Installation

To verify if the installation of Oracle Application Performance Monitoring is successful:

1. **Examine the APM Node.js Agent logs** located at `<APP_HOME>/oracle-apm/logs.`

   Note that the log files are not initially created, even if the installation of APM was successful. The log files get created and populated only when the application you are monitoring is running, and shows some activity which results in data depicted on the Oracle Application Performance Monitoring Web UI.

2. **Look for expected Directories and Files**

   Once your Node.js application is started, the following directories and files are available under the folder that contains your main module `(NODE_APP_HOME)`:

   -
     ```
     oracle-apm
     |___ config
            |____ oracle-apm-config.json
     |___ identities
            |____ AgentIdentity_server.json
     |___ logs
            |____ Agent_server.log
     ```
   - The `config` folder contains just one file - `oracle-apm-config.json` that has configuration information to connect to Oracle Management Cloud.
   - The `identities` folder contains the agent identity. This will have as many files as the number of main modules in your current `NODE_APP_HOME`. The files will be named as `AgentIdentity_<main_module>.json`. Thus if you have two main modules, `server.js` and `client.js`, in your current `NODE_APP_HOME`, this folder will have files `AgentIdentity_server.json` and `AgentIdentity_client.json`.

> **Note:**
>
> If this folder is deleted or if the files in this folder are deleted or edited, then, the application will take on a new agent identity on startup.

- The `logs` folder contains one log file per main module. Log files will be named in the format `Agent_<main_module>.log` . Thus, if you have two main modules, `server.js` and `client.js`, in your current `APP_HOME`, this folder will have files `Agent_server.log` and `Agent_client.log`.

3. **Look for successful Agent Start-up message** in `Agent_<main_module>.log`

   After a successful start-up, the following message should be found in the Agent log file:
   `INFO Agent with agentId (hex) {Agent_ID} successfully registered with the security service and retrieved its managed entity Id {Me_ID}`

4. **Check for Errors/Warnings in Agent log file**

   On starting the application for the first time after the deployment, the following messages are expected in the log:

   ```
   WARNING Unable to load json data from file <identity_file_name> :
   Error: ENOENT, no such file or directory <identity_file_name>

   WARNING Status 404 ,message: Not Found ,data received: Cannot
   recover the agent as the, given agent cannot be found in ODS,
   requestDetails: hostname:...
   ```

5. **Use the Oracle Application Performance Monitoring Web Console**

Configure APM Node.js Agents

You can use the APM Node.js Agent configuration tool to configure the APM agent in the user's application.

Prerequisites

- Ensure that the APM Node.js Agent is provisioned with Oracle Management Cloud successfully.
- Navigate to the Node.js application home and confirm that the following directory exists: `<application_home>/oracle-apm/config`.

  If the directory does not exist then run the following command to create it under the `<application_home>`: `mkdir -p oracle-apm/config`.

Initialize the agent configuration tool
To initialize the `oracle-apm` agent configuration tool in the directory of the Node.js application home, run the following command:

```
cd <application_home>
oracle-apm config init
```

As a result, two configuration files will be generated: `oracle-apm-config.json` and `url-normalizer-pattern.json` under the directory: `<application_home>/oracle-apm/config` which is the default configuration directory for the APM Node.js agent.

List the configuration properties

- To list the description of all the configuration properties, run the following command:

  ```
  oracle-apm config listall
  ```

- To list the current values of the configuration properties set in the Node.js application, run the following command:

  ```
  oracle-apm config list
  ```

Check a specific configuration property value
To check the value of a specific configuration property, run the following command:

```
oracle-apm config get <config_property_name>
```

Example:

```
oracle-apm config get injectionType
```

Update a specific configuration property value
To update the value of a specific configuration property, replacing the old configuration property value with a new one, run the following command:

```
oracle-apm config set <config_property_name> <config_property_value>
```

Example:

```
oracle-apm config set injectionType reference
```

Enable APM Node.js Agent

To enable APM Node.js Agent, the following line must be inserted as the first line in the monitored application's main module:

```
require('oracle-apm');
```

11

Install and Configure APM Ruby Agent

Topics:

- Requirements and Installation Instructions
- Verify the APM Ruby Agent Installation

Requirements and Installation Instructions

Prerequisites

- Supported versions of the application server:
  - Ruby 2.X and Rails 3.2
  - Ruby 2.X and Rails 4.2
  - Ruby 2.X and Rails 5.X
- If the application server you are running doesn't support TLS 1.2 security protocol, refer to My Oracle Support Doc ID 2703411.1 before proceeding with the APM agent installation.
- **Other considerations:**
  - The machine hosting the application server should be able to establish an HTTPS connection either directly or indirectly (using a proxy server or an Oracle Management Cloud gateway) to Oracle Management Cloud. For more information about Oracle Management Cloud gateway, see Install a Gateway.
  - The HTTPS connection must use TLS 1.2 security protocol.
  - The install user of APM Ruby Agent should be the same as the application server user.
  - The application server should have read and write permissions to the APM Ruby Agent `log` and `config` directories.

Deploy the Gateway (Optional)

A gateway is not a mandatory component while deploying Oracle Application Performance Monitoring. Use the gateway in the following scenarios:

- If you have an application server that does not support Transport Layer Security (TLS) protocol 1.2.
- If you have older versions of .NET IIS servers and Java Application Servers with JDK less than 1.7 (for example, Oracle WebLogic 10.3.6)

For instructions on how to deploy the gateway, see Installing a Gateway in *Installing and Managing Oracle Management Cloud Agents*.

Set the Gateway Variables (Optional)

Set the values for Gateway host and port.

- If you're using a Bash shell:

  ```
  export GW_HOST=<Gateway Host Name>
  export GW_PORT=<Gateway Port>
  ```

- If you're using a C shell:

  ```
  setenv GW_HOST "<Gateway Host Name>"
  setenv GW_PORT "<Gateway Port>"
  ```

If you are using more than one gateway, use the `--additional-gateways` option with the provisioning script.

Download the APM Ruby Agent Software

1. On the Oracle Management Cloud home page, click the Oracle Management Cloud Navigation icon on the top-left corner to view the Management Cloud navigation pane.
2. Select **Administration** and **Agents**.
3. On the Oracle Management Cloud Agents page, click the Action Menu on the top right corner of the page and select **Download Agents**. The Agent Software Download page is displayed.
4. From the **Agent Type** dropdown list, select **APM Agent**.
5. Click **APM Ruby Agent** to start downloading the installer to a local or shared directory in your data center.
6. Create a registration key that will be used during the time of installing a new agent. Oracle Application Performance Monitoring Cloud Service verifies this key before accepting any data sent by APM Ruby Agent deployed on your on-premises hosts. For more information about creating a registration key, see Manage Registration Keys in *Installing and Managing Oracle Management Cloud Agents*.
7. Click **Done** after the download is complete.
8. Extract the contents of the installer ZIP file.

Install and Provision APM Ruby Agent

After you have downloaded and extracted the installer, install and provision the APM Ruby Agent on your Rails application server.

To install the APM Ruby Agent:

1. Navigate to the directory where you downloaded and extracted the APM Ruby Agent software. This directory contains a ruby gem called **`oracle_apm-<version>.gem`**
2. **Install APM Ruby Agent Gem**

Install the gem to the Ruby installation your application will use:

```
gem install oracle_apm-<version>.gem
```

Example:

```
gem install oracle_apm-1.36.0.gem
```

3. **Provision Ruby Agent**

   a. The same download directory contains the file **`provision_ruby_agent.rb`**. Run the following command, using the Ruby installation that your application will use:

   ```
   ruby provision_ruby_agent.rb --help
   ```

   This should print out the script's help message, which lists all the parameters the script uses.

   b. Run the provisioning script as per your installation preference:

Installation Preference	Provisioning Script
Basic Installation	`ruby provision_ruby_agent.rb -d {rails_app_directory}`
With Gateway. If you are using more than one gateway, use the `--additional-gateways` option	`ruby provision_ruby_agent.rb -d {rails_app_directory} --gateway-host ${GW_HOST} --gateway-port ${GW_PORT} ruby provision_ruby_agent.rb -d {rails_app_directory} --gateway-host ${GW_HOST} --gateway-port ${GW_PORT} --additional-gateways https://{gw_host_2}:{gw_port_2},https://{gw_host_3}:{gw_port_3}`
Silent installation	`ruby provision_ruby_agent.rb -d {rails_app_directory}    --no-prompt --regkey-file {registration_key_file_path}`

Installation Preference	Provisioning Script
With Proxy	`ruby provision_ruby_agent.rb -d {rails_app_directory} --ph {http_proxy_host} --pp {http_proxy_port}`

- `-d` is the absolute path of the Ruby on Rails application directory. This is the directory that contains directories such as `config` and `log` for the application.
- `--ph {http_proxy_host}` (Optional) - the proxy server's host name
- `--pp {http_proxy_port}` (Optional) - the proxy server's port
- `--pu {http_proxy_user}` (Optional) - Authentication user for the http proxy if needed (if needed)
- `--ppw {http_proxy_password}` (Optional) - Authentication password for the http proxy (if needed)

c. When prompted, provide the value of the registration key that you've created or downloaded earlier. If you are running the provisioning script with the `--no-prompt` option, create a text file containing the value of the registration key, and provide the path to the file.
Example:

```
ruby provision_ruby_agent.rb -d sample_rails_app --no-prompt --
regkey-file regkey.txt
```

(where `regkey.txt` contains a single line with the registration key.)

4. **Enable the APM Ruby Agent for your application:** To have the APM Ruby Agent monitor your Ruby on Rails application, add the APM Ruby Agent gem to the application `Gemfile`.

   a. Update the Gemfile of the application by adding **gem 'oracle_apm'**. Here is an example:

```
source 'https://rubygems.org'
gem 'oracle_apm'
# Bundle edge Rails instead: gem 'rails', github: 'rails/rails'
gem 'rails', '4.2.6'
# Use sqlite3 as the database for Active Record
gem 'sqlite3'
...
```

5. Restart the Ruby on Rails Application. The next time your application is started, the APM Ruby Agent will start monitoring your application.

Provision APM Ruby Agent with a standalone installer

A standalone APM Agent installer is obtained when the Agent zip file is received via email, FTP or similar means (that is, when the agent zip file was not downloaded from an OMC server). To install using the standalone agent installer specify these additional parameters when running the provisioning script:

Oprion	Description
`--tenant-id`	The Oracle Management Cloud tenant name. You can get this value from the Agent Download page. • Script for v1 tenant: `ruby provision_ruby_agent.rb -d {rails_app_directory} --tenant-id {tenant} --omc-server-url {omc_server_url}` • Script for v4 tenant: `ruby provision_ruby_agent.rb -d {rails_app_directory} --tenant-id {service-tenant} --omc-server-url {omc_server_url}`
`--omc-server-url`	The URL of the Oracle Management Cloud server. If you are using gateways and have specificed `--gateway-host` and `--gateway-port`, you do not need `--omc-server-url`.

Verify the APM Ruby Agent Installation

You can verify if the deployment of the APM Ruby Agent is successful by examining the logs and verifying that the user interface displays the application.

APM Ruby agent logs are located in the Rails application log directory, within the `apm_agent` directory: `<rails_app>/log/apm_agent`.

As the Rails application starts up, the agent creates three log files:

- `agent.log`
- `agent_startup.log`
- `agent_status.log`

The `agent_startup.log` contains startup logs and will eventually log the following “Agent startup successfully completed”.

The `agent_status.log` file contains a summary of internal agent metrics showing the amount of traffic monitored, number of observations sent, and number of warnings or errors encountered among other data.

Verify Installation for multiple APM Ruby Agents:

If multiple servers are started for this Rails application, you can see a set of logs for each APM Ruby Agent. The first agent will look like the example above, and additional agents will have a number appended to the file name. For example:

- `agent_2.log`
- `agent_startup_2.log`
- `agent_status_2.log`

12

Install and Configure APM Agents on Containers

Topics:

APM Java Agent on Docker: Installation and Verification Instructions

Reference topic:

- Install and Configure an APM Java Agent

To install APM Java Agent in a Docker container:

1. Install the desired application server on the container.
2. Provision APM Java Agent in the container.

   ```
   ProvisionApmJavaAsAgent.sh -h do-not-use
   ```

   By default, the provisioning script can determine the machine hostname. By using the `-h do-not-use` parameter, you can override this default behavior and delay hostname determination until the application server starts up. Any value other than `do-not-use` will hardcode a hostname which it's typically undesirable for a container environment. An exception might be if there is a one-to-one mapping between apm agents with an assigned hostname and app server listen ports.
3. Modify the startup script of your application server to include the `-javaagent` property.
4. Build the image of the container.
5. Within the container, run the application and Application Performance Monitoring to monitor the performance of the application.

For an example of APM Java Agent on Docker using Kubernetes, see https://docs.oracle.com/en/solutions/monitor-applications-on-kubernetes/index.html.

APM Node.js Agent on Docker: Installation and Verification Instructions

Reference topics:

- Install and Configure APM Node.js Agent

To set up APM Node.js Agent within a Docker container, follow these additional steps apart from the ones listed in Agent Requirements and Installation Instructions.

1. After you have downloaded the APM Node.js Agent installation software and extracted it, copy the software into your Docker container.
2. Run the provisioning script to install the agent in the container. See Install and Provision APM Node.js Agent.
3. Copy the application files into the container. This is the application that will be monitored by Application Performance Monitoring.
4. Expose the required ports and start your Node.js application.
5. Build the image of the container.

   Here’s an example of a docker file ready to install APM Node.js Agent:

   ```
   COPY ./<Stage_DIR>/usr/src/provision

   ENV NODE_PATH /usr/local/lib/node_modules

   RUN /bin/bash ProvisionApmNodeAgent.sh -h <host>

   COPY ./<your application>/usr/src/<application folder>
   WORKDIR /usr/src/<application folder>

   EXPOSE 3000

   RUN npm -g list --depth=0

   CMD [ "npm", "start" ]
   ```

6. Spawn a new docker container:

   ```
   sudo docker run -h <host> -p <port>:3000 -d node-web-app
   ```

> **Note:**
>
> The parameter `-h` is essential for the hostname validation to succeed during provisioning the agent. This is required if your container is not able to resolve the `/etc/hosts` file. By default, the provisioning script can determine the machine hostname.
>
> By using the `-h do-not-use` parameter, you can override this default behavior.

Verifying the Installation

To verify the installation of the APM Node.js Agent in your container:

1. View the list of containers running on the host and get the ID of your container using this command:

   ```
   sudo docker ps
   ```

2. Login to the container using this command:

   ```
   sudo docker exec -it <container id>  /bin/bash
   ```

3. Navigate to the `oracle-apm` folder of your application. Check the folder containing the `container id` to locate the Agent's configuration and log folders. Verify the log contents are similar to the agent deployed on any other normal host.

APM Ruby Agent on Docker: Installation and Verification Instructions

Reference topic:

- Install and Configure APM Ruby Agent

Verifying the Installation

To verify the installation of the APM Ruby Agent in your container:

1. Check the `agent_startup.log` file to see if the APM Ruby Agent is running within the container:

   ```
   INFO <CONFIG> OS Container information: type=docker, identity=<long
   identity of the container>
   ```

2. In the `startup.log`, the Host discovery type has two new fields — `osContainerType` and `osContainerId`.

   Example

   ```
   25c4188 <2016-04-12T12:26:13:468> INFO <STARTUP>
   #<OracleAPM::HostInfo:0x0000000411f428 @hostName="myvm.jc",
   @osName="linux-gnu", @osVersion="", @architecture="x86_64-linux",
   @processorCount=4, @osContainerType="docker", @osContainerId="<long
   ```

```
identity of the container>", @agentIdKey="<agent id>=",
@agentVerKey="<agent version key>", @idKey="<identity key>",
@verKey="<version key>>
```

13

Troubleshoot the Deployment of Application Performance Monitoring

Troubleshoot APM Java Agent Deployment

Installation Issues

Connection Exception

During installation, the APM Java Agent issues several network requests.

Sometimes, on the first network request, you might get this below error:

```
Error while accessing the server: java .net.ConnectException: 
Connection timed out
ERROR: Agent configuration download failed
```

This error message indicates that the installer tried to issue a request to the OMC server or gateway, but did not receive any response. Usually the reason for that is that network traffic to the server should go through a proxy. The APM Java Agent Installer takes proxy settings from the following parameters: `-ph, -pp, -pt`. Note that the APM Java Agent Installer does not take into account environment variables like `HTTP_PROXY_HOST`, etc.

Troubleshooting — Agent Startup

APM Java Agent reports *Remote certificate is not trusted*

When installing the APM Java Agent, if the agent reports that the remote certificate is not trusted, create and add a remote certificate.

The remote certificate is the certificate presented to the agent, usually by Oracle Management Cloud, during the agent's attempt to establish an SSL connection. However, if the agent traffic goes through an intermediary, then, it is the certificate of the intermediary (for example, proxy) that might run into this error.

The certificate is included in the agent log information, and can be used to create a `.cer` file.

If the Java Agent's logs contain the SEVERE message **Remote certificate is not trusted**, AND if the Java agent's traffic goes through a proxy which presents a certificate not signed by a well-known certification authority, then, add the proxy's certificate(s) to the agent's trust list:

1. Edit your `AgentStartup.properties` file which it's located under `<APM agent install dir>/config` folder.
2. In the `pathToCertificates` property, add the full path to your proxy certificate `.cer` file.
3. Save the file.
4. Restart your application server.

If you don't have the proxy certificate handy, you could also copy it from the agent's log:

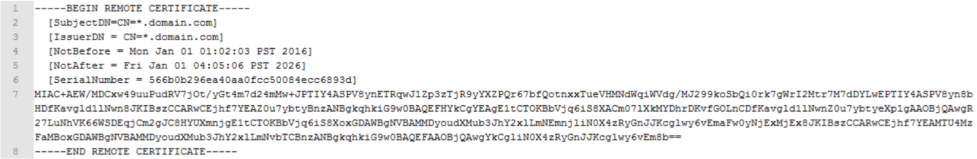
1. From the agent's log file, copy the block of lines starting with the line

   ```
   -----BEGIN REMOTE CERTIFICATE-----
   ```

   and ending with the line

   ```
   -----END REMOTE CERTIFICATE-----
   ```

2. Delete the word `REMOTE` (along with the trailing space) from both the `BEGIN` line and the `END` line.
3. Delete lines 2-6 (the information that describes the certificate).

   ```
   1  -----BEGIN REMOTE CERTIFICATE-----
   2    [SubjectDN=CN=*.domain.com]
   3    [IssuerDN = CN=*.domain.com]
   4    [NotBefore = Mon Jan 01 01:02:03 PST 2016]
   5    [NotAfter = Fri Jan 01 04:05:06 PST 2026]
   6    [SerialNumber = 566b0b296ea40aa0fcc50084ecc6893d]
   7  MIAC+AEW/MDCxw49uuPudRV7jOt/yGt4m7d24mMw+JPTIY4ASPV8ynETRqwJ1Zp3zTjR9yYXZPQr67bfQotnxxTueVHMNdWqiWVdg/MJ299koSbQi0rk7gWrI2Mtr7M7dDYLwEPTIY4ASPV8yn8b
      HDfKavgld1lNwn8JKIBszCCARwCEjhf7YEAZ0u7ybtyBnzANBgkqhkiG9w0BAQEFHYkCgYEAgE1tCTOKBbVjq6iS8XACm07lXkMYDhrDKvfGOLnCDfKavgld1lNwnZ0u7ybtyeXp1gAAOBjQAwgR
      27LuNhVK66WSDEqjCm2gJC8HYUXmnjgE1tCTOKBbVjq6iS8XoxGDAWBgNVBAMMDyoudXMub3JhY2xlLmNEmnjliN0X4zRyGnJJKcglwy6vEmaFw0yNjExMjEx8JKIBszCCARwCEjhf7YEAMTU4Mz
      FaMBoxGDAWBgNVBAMMDyoudXMub3JhY2xlLmNvbTCBnzANBgkqhkiG9w0BAQEFAAOBjQAwgYkCgliN0X4zRyGnJJKcglwy6vEm8b==
   8  -----END REMOTE CERTIFICATE-----
   ```

4. Save the file as a `.cer` file.

   The contents of the `.cer` file will look like this:

   ```
   -----BEGIN CERTIFICATE-----
   certificate base64 content
   -----END CERTIFICATE-----
   ```

Check for Agent trace in your Container log

If you have correctly provisioned the APM Agent on your container, the following traces should be visible in your container's log at startup.

Oracle WebLogic:

1. Check the container console log for the following lines:

   ```
   APM agent - preprocessing initialized
   APM agent - log directory location is /Users/JC/Oracle/wls12130/
   user_projects/domains/agentDomain/apmagent/logs/AdminServer
   ```

2. Check if the agent log files (For example, `AgentStartup.log`) are being created.

If the above tasks are not being performed, verify your agent installation. Check if the APM Agent is added to the server startup script.

For the server JVM to start with the APM Agent runtime, the container startup script should typically contain something similar to this:

```
JAVA_OPTIONS="${JAVA_OPTIONS} -javaagent:$DOMAIN_HOME/apmagent/lib/
system/ApmAgentInstrumentation.jar"
```

If the above lines are missing, add them to the container startup script and restart.

Apache TomCat:

1. Check if the file `catalina.out` contains lines similar to the following:

   ```
   APM agent - preprocessing initialized
   APM agent - log directory location is /scratch/tomcat/apache-tomee-
   plus-1.7.2/apmagent/logs/tomcat_instance
   ```

2. Ensure that the `-javaagent` parameter is passed at startup. To do that, check if the file `catalina.sh` contains the following lines:

   ```
   CATALINA_OPTS="${CATALINA_OPTS} -javaagent:${CATALINA_HOME}/
   apmagent/lib/system/ApmAgentInstrumentation.jar"
   ```

Container does not start

The APM Agent waits for the container to be ready to start its initialization sequence. Although rare, it could be that the container is never ready, or takes a long time to reach a ready state. You can check this in the `AgentStartup.log` file.

```
0xb<2015-06-25T14:08:10.169+0200> INFO <STARTUP> The container is still
initializing and is therefore not ready for observation processing
0xb<2015-06-25T14:08:10.169+0200> INFO <STARTUP> Another message will
get logged once thecontainer is ready and agent initialization can
start
0xb<2015-06-25T14:08:10.169+0200> INFO <STARTUP> If you do not see such
a message, pleasecheck the container startup logs.
```

These traces will be followed by a message explaining the container status. If there is any container startup issue, the container output logs (and not the agent logs) will contain information about the problem. If the container waits too long to start, the agent fails its initialization sequence.

By default, the APM Agent waits a maximum of 10 minutes for the container to be ready. You can increase this time by changing the system property `oracle.apmaas.agent.container.startupWaitTime` in the AgentStartup.properties file. This property is specified in seconds.

SSL connection fails

If the remote server certificate is not trusted, an explicit message gets logged, along with the remote certificate content:

```
0xb<2015-06-25T15:17:04.968+0200> INFO <common.agentToEngine.transport>
Read custom certificate from /var/opt/ORCLemaas/sec/cert.cer
0xb<2015-06-25T15:17:04.968+0200> WARNING
<common.agentToEngine.transport> Remote certificate is not trusted
[SubjectDN=CN=*.example.com]
[IssuerDN = CN=*.example.com]
[NotBefore = Thu May 21 22:43:40 CEST 2015]
[NotAfter = Sun May 18 22:43:40 CEST 2025]
[SerialNumber = b5d3145ced001866f475ecdde44cbd58] <Ref:
GBIZRBMPLYN3AVXZWNJDUTSJBINCBRQI>
0xb<2015-06-25T15:17:04.969+0200> INFO <common.agentToEngine.transport>

-----BEGIN CERTIFICATE-----
MIIBszCCARwCEQC10xRc7QAYZvR17N3kTL1YMA0GCSqGSIb3DQEBCwUAMBoxGDAWBgNVBAMM
DyoudXMub3JhY2xlLmNvbTAeFw0xNTA1MjEyMDQzNDBaFw0yNTA1
MTgyMDQzNDBaMBoxGDAWBgNVBAMMDyoudXMub3JhY2xlLmNvbTCBnzANBgkqhkiG9w0BAQEF
AAOBjQAwgYkCgYEAi/Y/58x4NGOeToiHn7b+T/QjpG7ZutA1by3x
f71Y8qvvFDO7AD1VsrG464YkauinR/DQOCovhvXwyYG/
HnRE2SCVS9nOTotve37QaD92Bs6Mt0Gku1/2X3HYa6JxjQ+l4VwmhItYFEMPMfe0ZHtQpz+4
4psQxOS1
rT402EIA0DsCAwEAATANBgkqhkiG9w0BAQsFAAOBgQAqBSxip3+yjX3j5gk4OButP8b9S3Qb
l1pR1KwWx22NCuSW7a8KL3C+BPQPtR0YpxxgMC4F/VOGkEkOBrjY
mG7fULYU8f7ab8ck6oHHdl0CPztp/
mxRDpWSizBNKlUCSThxKqvSVEtEZrsh5zhn0VofiRlbZwZBWu4C5ObbjvZ8iw==
-----END CERTIFICATE-----
```

This enables you to compare the remote certificate with the trusted certificate(s) used by the APM Agent. If you can trust the remote certificate, and want to bypass the trust check, define the property `oracle.apmaas.common.trustRemoteSSLHost`, and set it to `true` in the `AgentStartup.properties` file.

Unsupported Cryptographic Protocol

OMC uses TLS 1.2 for SSL cryptographic protocol. An INFO message is logged with the cryptographic protocols support by the current JVM.

```
0xb<2018-02-21T19:49:53.007+0000> INFO <common.agentToEngine.transport>
Supported crypto protocols: [TLSv1]
```

If supported protocol does not have TLSv1.2, and Oracle Gateway is not used, the following or a similar message is logged.

```
    0xb<2018-02-21T19:49:54.852+0000> WARNING
<common.agentToEngine.transport> Error connecting to https://
<Oracle_cloud_host_name>/static/regmanager/agents

    Unable to establish SSL connection to destination server. The
current Java version may not support TLS 1.2 cryptographic protocol.
```

```
    (set -Djavax.net.debug=ssl to confirm this, since this exception
could happen for other reasons too)
    If confirmed, the solution is to either update Java to a version
that supports TLS 1.2 (Java 1.6.0_121 or later) OR
    send the agent traffic through an Oracle Gateway Agent

    <javax.net.ssl.SSLHandshakeException: Received fatal alert:
handshake_failure> <Ref: LWT7DYU6HP2DVWY4EQWJX7ZZIA5SML5A>
```

There are 2 ways to allow Agent to communicate with OMC:

1. Upgrade Java to at least Java 1.6.0_b121
2. Route Agent traffic through an Oracle Gateway Agent.

Communication fails

If there is any communication failure, it gets logged as a WARNING, for example:

```
0xb<2015-06-25T14:08:26.407+0200> WARNING <agentToEngine.emaas>
Response NOT OK ServiceInfo: RegistryService - agent: null ,tenant:
apm_testtenantx1 serviceName: DataReceiver.storage ,version: null
Transport info: HTTP method: GET ,URL:https://abc.com:4443/registry/
instances?status=UP&serviceName=DataReceiver.storage ,response status:
503 ,response headers: null=HTTP/1.1 503 Service Unavailable ,
X-92eeb115-fa68-449e-9df7-c2d3ec508ca0-reroute= , Content-Language=en ,
Access-Control-Allow-Headers=Origin, X-Requested-With, Content-Type,
Accept,X-USER-IDENTITY-DOMAIN-NAME,Authorization,x-sso-client , X-
ORACLE-DMS-ECID=0056LPv4ofAEWNI_IpWByf0002M^0000_d , Access-Control-
Allow-Origin=* , Date=Thu, 25 Jun 2015 12:08:26 GMT , Content-
Length=0 , Set-
Cookie=_WL_AUTHCOOKIE_EMCS_JSESSIONID=lNpT3Z9FK9OqhZ2lSqQS; path=/
registry; secure; HttpOnly
EMCS_JSESSIONID=WUYqnnu9LFdZXmiWulLknPsGK0pIP_T1QaFgvNvPol0Jwre8OJtF!-11
21177946; expires=Thu, 25-Jun-2015 12:08:36 GMT; path=/registry;
secure; HttpOnly , Access-Control-Allow-Methods=GET, OPTIONS, HEAD ,
Connection=close , Content-Type=application/json; qs=1 <Ref:
IQWJEECVYYIRYOFG5HDJUJW43RYOTU4W>
```

Also, additional messages that are more user-friendly, might get logged:

```
0xb<2015-06-25T14:08:26.408+0200> WARNING <PROCESSING> The registry
service replied to the agent with an error code. Please
check the log and service status for more information. <Ref:
74RXY6IPUAFSBWSROR2VXMEGJRI7HEIU>
0xb<2015-06-25T14:08:26.408+0200> WARNING <PROCESSING> The agent
could not get a data receiver address from the
registry service location 'https://abc.com:4443/registry/'. <Ref:
Z5GXIKFE6C52BKKWHGXOQYQMCGCEYQB3>
```

If the agent cannot connect to the remote services, the agent cannot complete its initialization and will keep retrying until communication channels are working.

```
0x37<2015-06-25T15:17:04.983+0200> WARNING <STARTUP> No managed entity
Id could be retrieved from the target model and security service.
Since the agent needs a meId to be operational, the agent will
now keep trying to lookup a value. A message will get logged
once it succeeds. No traffic will get monitored until then. <Ref:
MX5B6MBD4UJIVP2JDFJUNU65AHHA4FLE>
0xb<2015-06-25T15:17:05.984+0200> INFO <STARTUP> Agent startup is
waiting for the full initialization of its communications with remote
services, and successful delivery of its core observations. Another
message will get logged when the agent is operational and its
initialization complete.
0x37<2015-06-25T15:22:30.751+0200> INFO <bootstrap> Agent still trying
to lookup an managed entity id value from the target model and security
service. Agent still not operational.
0x37<2015-06-25T15:28:05.325+0200> INFO <bootstrap> Agent still trying
to lookup an managed entity id value from the target model and security
service. Agent still not operational.
0x37<2015-06-25T15:33:40.481+0200> INFO <bootstrap> Agent still trying
to lookup an managed entity id value from the target model and security
service. Agent still not operational.
0x37<2015-06-25T15:39:17.542+0200> INFO <bootstrap> Successfully looked
up a managed entity Id: 63EB5524C11743EEA47C09C3CBB94CB6
0x37<2015-06-25T15:39:17.542+0200> INFO <bootstrap> Agent core
discovery observations successfully sent
0x37<2015-06-25T15:39:17.542+0200> INFO <bootstrap> Agent startup
successfully completed - the agent is now operational and monitoring
traffic
```

RegistrationKey not correct

The agent might start by trying to get its existing managed entity ID, assuming that it already registered during a prior startup (**bold** INFO log below). If it fails or if the agent just notices that it needs to perform an initial registration, it tries to register. A failure to register (read **INFO** below), is the sign of an invalid registration key.

```
0xb<2015-06-26T15:20:09.134+0200> WARNING <agentToEngine.emaas>
Response NOT OK ServiceInfo: SecurityServiceRegister
- agent: null ,tenant: apm_testtenantx1 ,registrationKey:
pTaz5UiPcLbnKvxyVayD4V ,entityName: null ,clientId: null Transport
info: HTTP method: POST ,URL: https://abc.com:4443/microservice/
96122404-13cf-46cd-a9fb-afdeb4a1df21/agents ,response status:
404 ,response headers: null=HTTP/1.1 404 Not Found , Content-
Language=en , X-ORACLE-DMS-ECID=0056MiPMKrWDGfQ_I_T4if0006dN00003u ,
Transfer-Encoding=chunked , Date=Fri, 26 Jun 2015 13:20:08
GMT , Keep-Alive=timeout=5, max=100 , Connection=Keep-Alive , Content-
Type=application/json Transport content: Received status 404
from dependent service http://abc.com:7001/clientservices-persistence/
registration = 404, Check service logs for string =CS-1435324808534 for
more details <Ref: 37DZJ3LK64ACGZOWPFFBNFFPFYJ5UXSO>
0xb<2015-06-26T15:20:09.135+0200> INFO <agentToEngine.emaas> The agent
could not get a managed entity ID value from the Security Service
(returned status: 404) - the agent might not be registered. Trying to
```

```
register now.
0x1c<2015-06-26T15:20:09.325+0200> INFO <ACTION.JAXWS> JAXWS probe
adding server side handler
0xb<2015-06-26T15:20:09.565+0200> WARNING <agentToEngine.emaas>
Response NOT OK ServiceInfo: SecurityServiceRegister
- agent: null ,tenant: apm_testtenantx1 ,registrationKey:
pTaz5UiPcLbnKvxyVayD4V ,entityName: null ,clientId: null
Transport info: HTTP method: POST ,URL: https://abc.com:4443/
microservice/96122404-13cf-46cd-a9fb-afdeb4a1df21/agents ,response
status: 500 ,response headers: null=HTTP/1.1 500
Internal Server Error , Content-Language=en , X-ORACLE-DMS-
ECID=0056MiPPlRxDGfQ_I_T4if0006dN00003v , Transfer-Encoding=chunked ,
Date=Fri, 26 Jun 2015 13:20:09 GMT , Connection=close , Content-
Type=application/json Transport content: Received status 400 from
dependent service http://abc.com:7001/targetmodel/api/v1/data/mes,
Check service logs for string =CS-1435324809465 for more details <Ref:
NEXSTWGNGQTZNVFAUTP4DPQNIYILIXUB>
0xb<2015-06-26T15:20:09.565+0200> INFO <agentToEngine.emaas>
Registration attempt to the Security Service did not return a managed
entity ID. Will keep retrying.
0xb<2015-06-26T15:20:09.565+0200> WARNING <PROCESSING> The agent failed
getting a managed identity Id - please check the logs for additional
information. <Ref: P223WCKDF6KETCRSVCFVTWGS6ZO2MW2Q>
```

The registration key is specified in the `AgentStartup.properties` file, and you can change its value if the registration key is not correct.

Invalid credentials

If credentials to authenticate OMC are not correct, a transport message WARNING gets logged, and HTTP 401 status is returned. Depending on your setup, the credentials will be either located within a wallet or encrypted within the `AgentHttpBasic.properties` file.

```
0xb<2015-06-27T06:38:49.697+0200> WARNING <agentToEngine.emaas>
Http credentials were not authorized to access
the service. Will attempt to read credentials
again ServiceInfo: RegistryService - agent: null ,tenant:
apm_testtenantx1 serviceName: DataReceiver.storage ,version: null
Transport info: HTTP method: GET ,URL:https://abc.com:4443/registry/
instances?status=UP&serviceName=DataReceiver.storage ,response status:
401 ,response headers: null=HTTP/1.1 401 Unauthorized , Content-
Language=en , WWW-Authenticate=Basic realm="weblogic" , Date=Sat, 27
Jun 2015 04:38:49 GMT , Content-Length=1468 , Keep-Alive=timeout=5,
max=100 , Connection=Keep-Alive , Content-Type=text/html; charset=UTF-8
<Ref: DPKGO5GY2GMNIMOOV7FLSKTMQCLU2FGU>
0xb<2015-06-27T06:38:49.699+0200> WARNING <PROCESSING> The agent could
not authenticate to the registry service. Make sure that the
credentials specified are correct. There is no need to restart the
container if you update the agent credentials as the agent will keep
trying to connect until it succeeds, using the more recent set of
available credentials. <Ref: XD2BEOVFLOGTE5XK6PH3HFBNUPWYX6EF>
```

Note that the remote service might have a lockout period. Fixing credentials to the correct values might not be sufficient to reconnect immediately. Wait for the lockout period to expire before the agent can reconnect.

OSGi (Open Services Gateway initiative) property setting

If the application you would like to monitor has a dependency on OSGi, make these manual settings to ensure proper framework boot delegation so that the application that is being monitored does not break.

- **WebLogic Server**: On your WebLogic Server, ensure the monitored OSGi framework instances have the Java system property, `oracle.apmaas.*` added to the `Framework Boot delegation` property as follows:

  ```
  -Dorg.osgi.framework.bootdelegation=oracle.apmaas.*
  ```

  Refer to the WebLogic documentation on ways to change the WebLogic OSGi settings.
- **Atlassian JIRA Felix OSGi container (Tomcat)**: Add the following option to the JIRA container's startup options:

  ```
  -
  Datlassian.org.osgi.framework.bootdelegation=oracle.apmaas.*,sun.*,o
  rg.apache.xerces,org.apache.xerces.*,org.apache.naming,org.apache.na
  ming.*,org.apache.catalina,org.apache.catalina.*
  ```

Logs not created

If APM Agent logs are not created even when the application is running, check if the `-javaagent` option for `ApmAgentInstrumentation.jar` was added correctly to the server startup command.

Security Access errors while starting APM Java Agent

If you run the APM Java Agent with a Java security manager and see an error message with the following content:

`java.security.AccessControlException: access denied()`

or

`access denied()`

add the following block to the Java security policy file:

```
grant codeBase "file:<path_to>/apmagent/-" { permission
java.security.AllPermission; };
```

Unable to open TomEE service during installation

While configuring APM Agent as a Windows Service on TomEE, and you run `TomEE.exe` and see the following error:

The specified service does not exist as an installed service. Unable to open the service 'TomEE'.

This means that the Windows service name is not the default value, that is, TomEE.

Workaround: Specify the exact service name you have provided for TomEE in the command prompt:

```
TomEE.exe//ES//<service_name>
```

Unable to get OAuth Token from IDCS Server

If the agent startup log shows that it cannot get the initial OAuth authentication token, preceded by a warning showing a failure to reach the IDCS server, check to ensure that there is no firewall blocking access to the IDCS server. If there is a firewall, you will need to allow access to the IDCS server.

Trust Manager or Trust Anchor related errors

If you see any of the below errors in the APM logs:

```
javax.net.ssl.SSLException: java.lang.RuntimeException: Unexpected
error:
      java.security.InvalidAlgorithmParameterException: the
trustAnchors parameter must be
      non-empty
```

then, the trust store may be invalid.

Workaround: Check if the property `javax.net.ssl.trustStore` is being passed. If yes, check that the full path to the trust store is specified. If yes, check the trust store's validity using the JDK **Keytool** utility.

Could not Generate DH Keypair

If APM Java agent provisioning fails with an SSL exception with the error *Could not generate DH keypair,* this issue could be due to a JDK bug that has been fixed. Check the version of your JDK, and update your JDK to a patch level that resolves this issue (for example, this problem happens with JDK 1.7.0_65, and updating to 1.7.0_201 fixes the issue).

Websphere Application Server doesn't start after uninstalling APM Agent

If after uninstalling APM Agent from your Websphere Application Server, the application server does not start, follow these steps to check if `-javaagent` is specified correctly:

1. In an editor, open the file `$WAS_HOME/config/cells/<celll-name>/nodes/<node-name>/servers/<server-name>/server.xml`.
2. Search for `genericJvmArguments` and look for the `-javaagent` option.
3. Remove the `-javaagent` option, and save the file.
4. Replace the current `server.policy` startup script of your WebSphere server with the original one you had before installing the APM agent.
5. Restart the Websphere server.

Oracle Forms monitoring is not working after deploying APM Java Agent

If Oracle Forms monitoring is not working after deploying APM Java Agent, you can check the log file: `AgentErrors.log` and look for the following errors:

- Connect timed out message:

  ```
  Unable to POST to collector due to IOException: connect timed out
  Exception in thread "main" java.net.SocketTimeoutException: connect
  timed out
  ```

  This error message indicates that you might need a proxy server. To fix it, add the following parameters to the file `AgentStartup.properties` which it's located under `<APM agent install dir>/config` folder.

  ```
  oracle.apmaas.common.proxyHost = my-proxy.example.com
  oracle.apmaas.common.proxyPort = 80
  ```

- Handshake failure message:

  ```
  Unable to POST to collector due to IOException: Received fatal
  alert: handshake_failure
  Exception in thread "main" javax.net.ssl.SSLHandshakeException:
  Received fatal alert: handshake_failure
  ```

  This error message indicates that you might be running an older JDK version. Collector needs you to use TLSv1.1 or TLSv1.2 to connect it. Please ensure that your JDK support any of these TLS versions and then set it up doing the following:

  ```
   oracle.apmaas.agent.forms.tlsProtocol = TLSv1.2
  ```

- No valid certificate message:

  ```
  Unable to POST to collector due to IOException:
  java.security.cert.CertificateException: No valid server
  certificate found
  Exception in thread "main" javax.net.ssl.SSLHandshakeException:
  java.security.cert.CertificateException: No valid server
  certificate found
  ```

  This error message indicates that you do not have the correct collector certificate in the `certificates` folder. Try downloading the certificate from the collector and place it in the folder: `apmagent/config/certificates`.

  You can download the certificate via a browser by navigating to the collector URL and saving it as a DER encoded binary file.

- If Forms Name is not configured then the `AgentErrors.log` file reports the following error:

  ```
  0x4b<2019-11-27T05:59:50.304-0800> WARNING <HANDLER.FORMS>
  <005a5Hcz7E2Dg^0_rx9DiY00057l000kDc>
  FormWindow Message detected  without INDEX_FORM_MODULE property.
  ```

```
Please enable the Oracle Forms to  send forms name by setting its 
property
FORMS_RUEI_SEND_FORM_NAME=TRUE  in your Oracle Forms environment. 
This can be set in file
'default.env'  by your Forms Administrator. Please refer to APM 
agent documentation for  more
info on this configuration  <Ref:  ZHZDUUVYBRU7XVFZG6R5LF64Q3JMJCAF>
```

Spring Boot 2.2 with Tomcat is not being detected by APM Agent

The APM Agent is not visible in the console and the APM Agent log has the following message:

```
<2020-05-19T21:11:56.478+0000> SEVERE <STARTUP> Failed to get 
container information after waiting for 600 seconds <Ref: 
WZ7PXNQDOUY5PK4YALTWGAZ4UHVLWUKG>
<2020-05-19T21:11:56.487+0000> SEVERE <STARTUP> Agent failed to start 
<Ref: ZQNZ2FKNIJJCZTD4DUWDCAXKKNE4W3ZS>
```

To resolve it, try the following:

- If you are using Spring Boot 2.2 with Tomcat, enter the following two properties in the application.properties file:

  ```
  spring.jmx.enabled=true
  server.tomcat.mbeanregistry.enabled=true
  ```

  By default, the application.properties file is located inside the spring-boot app executable jar, under the BOOT-INF/classes directory. Spring boot allows you to have many locations for this file and multiple formats. For more details, see Spring Boot Application Property Files.

  For information about the Spring Boot 2.2, see Spring Boot 2.2 Release Notes.

- If you are using other Application Server, you can force the APM Agent to use a specific application server name by setting the custom value provided using the Custom AppServer feature.

  To activate the Custom AppServer feature, do the following:

  1. Set up the following Java system property in the Java startup argument:

     ```
     oracle.apmaas.agent.custom.appserver.name
     ```

     When oracle.apmaas.agent.custom.appserver.name property is specified, Java APM Agent will look for the custom-appserver.properties file in the server config directory such as apmagent/config/<dir_name>/custom-appserver.properties.

  2. Create the custom-appserver.properties file if it doesn't already exist.

     If custom-appserver.properties file exists, Java APM Agent uses it to populate the app server container details, and discovers the app server based on the provided details.

If `custom-appserver.properties` file does not exist, Java APM Agent assumes this is a J2SE application with default properties and a J2SE app server will be discovered.

The `custom-appserver.properties` file should be created manually before Java APM Agent is run.

custom-appserver.properties file

The `custom-appserver.properties` file has the following properties:

Property Name	Defaults for J2SE	Defaults fro custom-appserver.properties file
type	"Java SE"	No default
name	System.getProperty("oracle.apmaas.agent.custom.appserver.name") + "(" + System.getProperty("user.dir") + ")"	No default
version	RuntimeMXBean.getSpecVersion()	
vendor	RuntimeMXBean.getVmVendor()	
path	System.getProperty("user.dir")	System.getProperty("user.dir")
ports		
sslPorts		

Sample of `custom-appserver.properties` file

```
type=Jetty
name=My Jetty Sandbox
version=9.2.5
vendor=Eclipse
ports=8080
sslPorts=8443
```

Troubleshoot APM Node.js Agent Deployment

Installation Issues

Certificate log errors while deploying APM Node.js Agent

If you see a warning message in the logs with the following content, then delete the file `emcs.cer` from `NODE_PATH/oracle_apm/data/`.

Warning:

```
Failed to call hostname:
<registry_server>, port: <path>l, path: /registry/instances?
status=UP&serviceName=SecurityService&version=1.0%2B, method: GET,
header: X-USER-IDENTITY-DOMAIN-NAME=<tenant_id>, header: Content-
Type=application/json ,failure message: certificate not trusted
```

Unable to get OAuth Token from IDCS Server

If the agent startup log shows that it cannot get the initial OAuth authentication token, preceded by a warning showing a failure to reach the IDCS server, check to ensure that there is no firewall blocking access to the IDCS server. If there is a firewall, you will need to allow access to the IDCS server.

Troubleshoot APM .NET Agent Deployment

Installation Issues

Cannot find the folder *c:\ProgramData*

If you cannot find the `ProgramData` folder, do one of the following:

1. Check if the `ProgramData` folder is in the Hidden Folders.
2. Open the command prompt, and input `set`, and then check for the environment variable `ProgramData`.

```
C:\Users\mimu>set
...
ProgramData=C:\ProgramData
ProgramFiles=C:\Program Files
ProgramFiles(x86)=C:\Program Files (x86)
ProgramW6432=C:\Program Files
SystemRoot=C:\windows
TEMP=C:\Users\mimu\AppData\Local\Temp
TMP=C:\Users\mimu\AppData\Local\Temp
...
```

Error while adding collection entry

The following error is displayed while trying to add collection entry with unique key attribute `name` set to `Oracle APM IIS Module`.

Error: Cannot add duplicate collection entry of type 'add' with unique key attribute 'name' set to 'Oracle APM IIS Module'

Here’s how to fix this:

1. Open Windows file explorer, and browse to `C:\Windows\System32\inetsrv\config`.
2. Make a backup copy of the file `applicationHost.config`.
3. Open the `applicationHost.config` with Notepad. (We recommend the use of Notepad here.)
4. Search for *Oracle APM IIS Module*, and remove the duplicate entries.
5. Save the `applicationHost.config` file.

Try adding the collection entry again.

Troubleshooting — Agent Startup

Ensure that the Instrumentation is working

You can check if the instrumentation is working by following these steps:

1. Open command prompt and check the environment variables:

   ```
   C:\>set cor
   COR_ENABLE_PROFILING=1
   COR_PROFILER=oracle.apmAgent
   ```

2. Check the environment variables of **w3wp.exe** process.
   a. Reset your IIS server.
   b. In the Process Explorer, open the application being monitored.
   c. Locate the **w3wp.exe** process, and view the Properties.
   d. Open the Environment tab, and check for the environment variables.

3. Check `registry: regedit`

   ```
   HKEY_CLASSES_ROOT\oracle.apmAgent\CLSID
           (Default)={15837040-7EC4-4B70-AEB0-EB1ADC26960D}
   HKEY_CLASSES_ROOT\CLSID\{15837040-7EC4-4B70-AEB0-EB1ADC26960D}
   \InprocServer32
           (Default)=C:\Program Files\Oracle APM .NET
   Agent\OracleAPMInstrumenter64.dll
   HKEY_CLASSES_ROOT\Wow6432Node\CLSID\{15837040-7EC4-4B70-AEB0-
   EB1ADC26960D}\InprocServer32
           (Default)=C:\Program Files (x86)\Oracle APM .NET
   Agent\OracleAPMInstrumenter32.dll
   HKEY_LOCAL_MACHINE\SOFTWARE\Classes\CLSID\{15837040-7EC4-4B70-AEB0-
   EB1ADC26960D}\InprocServer32
           (Default)=C:\Program Files\Oracle APM .NET
   Agent\OracleAPMInstrumenter64.dll
   HKEY_LOCAL_MACHINE\SOFTWARE\Classes\Wow6432Node\CLSID\{15837040-7EC4
   -4B70-AEB0-EB1ADC26960D}\InprocServer32
           (Default)=C:\Program Files (x86)\Oracle APM .NET
   Agent\OracleAPMInstrumenter32.dll
   HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Classes\CLSID\{15837040-7EC4
   -4B70-AEB0-EB1ADC26960D}\InprocServer32
           (Default)=C:\Program Files (x86)\Oracle APM .NET
   Agent\OracleAPMInstrumenter32.dll
   ```

4. Check the Windows event viewer to see if there are any errors regarding CLR profiling.

5. Use a process exploration tool to check if `OracleAPMInstrumenter64.dll` or `OracleAPMInstrumenter32.dll` is loaded.

   Example: You can use Procexp. Go to *Find Module* and search for **OracleAPM**.

6. You can install a simple sample on the problematic environment and see if the `OracleAPMInstrumenter64.dll` can be loaded.

HTTP Request Error

If you find a HTTP request error after enabling End User Monitoring, disable the offending URLs through configuration. See Disabling Monitoring by APM .Net Agent, and use the *Disable specific URLs* option.

Intermittent HTTP Request Failures

If there are intermittent HTTP request failures after installing the APM .NET Agent, check the performance monitor and find out the resource bottleneck. If there are queued requests, that could mean that the threadpool was initialized with less threads than needed. In this case, they can set up thread pool to have sufficient initial worker threads.

Edit the `machine.config` files with below configuration:

```
<configuration>
        "<system.web>"
      <processModel autoConfig="true" minWorkerThreads="30" />
      </system.web>
</configuration>
```

The `machine.config` files include:

- C:\Windows\Microsoft.NET\Framework64\v4.0.30319\Config\machine.config
- C:\Windows\Microsoft.NET\Framework\v4.0.30319\Config\machine.config

Unable to get OAuth Token from IDCS Server

If the agent startup log shows that it cannot get the initial OAuth authentication token, preceded by a warning showing a failure to reach the IDCS server, check to ensure that there is no firewall blocking access to the IDCS server. If there is a firewall, you will need to allow access to the IDCS server.

Error indicating IIS is not installed

While installing the APM .Net Agent, if you see an error that says IIS is not installed, follow these steps:

1. Through the File Explorer, access this file: `c:\windows\system32\inetsrv\config\applicationHost.config`
2. In the confirmation box, confirm that you want to access the folder that contains the above file.
3. Click **OK**. Ensure that the file `applicationHost.config` exists, and then try to run the APM .Net installer again.

Windows Event Viewer reporting Event ID 1022 as an error

You might see Event ID 1022 flagged as an error in Windows Event Viewer. This event is signaling that the APM Agent profiler loading failed. This event entry is expected and can be ignored.

By design, the APM Agent decides to only monitor IIS worker processes, and not to monitor other processes. Whenever the APM Agent decides not to monitor a process

- therefore any non IIS worker process - the APM Agent profiler is not loaded and this event gets logged in Event Viewer.

APM .NET Agent might not work due to conflicts with .NET Profile API

APM .NET Agent might not work correctly if other software that uses *.NET Profile API* is installed on the same machine. In most cases, this would be another monitoring tool such as Microsoft Monitoring Agent, AppDynamics, New Relic, etc.

This conflict usually results in the lack of reported events in the APM UI despite the fact that APM .NET Agent is successfully deployed, connected and reporting. In this situation it is advisable to use *Microsoft Process Explorer* (https://docs.microsoft.com/en-us/sysinternals/downloads/process-explorer) to examine the environment variables of one of the `w3wp.exe` processes. In case `COR_PROFILER` variable is defined and its value is not `oracle.apmAgent`, the conflict described above is likely to occur.

The resolution is to uninstall both APM .NET Agent and the conflicting software, reboot the machine and then reinstall APM .NET Agent.

Troubleshoot APM Ruby Agent Deployment

Installation Issues

Check for the Startup files

To check if the APM Ruby Agent is properly installed and started, locate the agent's **log** directory and look for the `agent_startup.log` file. If the agent is installed for a Rack-base application like Rails, the log directory will be located in the directory `log/apm_agent` under the application root directory. If the agent home directory was specified (`ORACLE_APM_RUBY_AGENT_HOME`), the log directory will be located within the agent home directory, under `logs/<appname>`.

If the `agent_startup.log` file is not present, this means that the agent startup sequence was not triggered or that an early error happened. You might want to check the following:

1. **Check if Gem is installed**:

   Verify that the `oracle_apm gem` is present in the gem library.

   ```
   $ gem list oracle_apm
   ```

   should display the agent gem version you expect to run. If it is not present, install it using the gem install command. If more than one version is present, make sure to specify the right version in the Gemfile definition.

2. **Check the Gem definition**:

   If the gem is present in the gem library, ensure that it is correctly referenced in the application Gemfile. For example: `gem 'oracle_apm', '~> 1.27.1'`

   If there is no Gemfile for the Ruby process to monitor, ensure that the `oracle_apm` gem gets loaded. For example, adding a `require ‘oracle_apm’` statement to the bootstrap sequence of your application.

3. **Agent home**

If the log directory or file cannot be found under the Rails log directory, check that the ORACLE_APM_RUBY_AGENT_HOME is defined. This is where the log would be located.

4. **Console log**

   You can force the agent bootstrap logs to be logged to the console (stdout) by setting the environment property ORACLE_APM_RUBY_AGENT_CONSOLE_LOG to any value (example: true). If you do not see anything logged as you restart your application, this means that the gem was not loaded. Check that the gem is correctly defined in the Gemfile or properly loaded by your application.

Initialization Process

As the agent starts up, it logs in detail, its initialization sequence in the log file named agent_startup.log. Once the agent is initialized and ready to monitor traffic, it logs the following line:

```
INFO <STARTUP> Agent startup successfully completed - the agent is now
operational and monitoring traffic
```

1. **Configuration**

   If there is a typo in the YAML agent configuration file, agent startup might fail. Because configuration is processed early on during the agent startup, you might need to use the environment property ORACLE_APM_RUBY_AGENT_CONSOLE_LOG mentioned above, to force initial logging to go to the console. If there is any typo in the configuration file, error parsing the YAML configuration file will indicate which line caused the problem.

2. **SSL**

   The agent communicates securely to OMC through SSL. If there is any problem establishing a secured connection, reasons will get logged in the startup log file. If you need to add a new certificate as a trusted server certificate, you can add the certificate in the agent configuration directory, in DER or PEM format with the .cer file extension. Upon restart, the agent will use this additional certificate as a trusted certificate.

Unable to get OAuth Token from IDCS Server

If the agent startup log shows that it cannot get the initial OAuth authentication token, preceded by a warning showing a failure to reach the IDCS server, check to ensure that there is no firewall blocking access to the IDCS server. If there is a firewall, you will need to allow access to the IDCS server.

14

Set Up End User Monitoring

After you have deployed the APM Agent, follow these steps to set up End User Monitoring.

Enable and Configure End User Monitoring

You can enable End User Monitoring and configure the type of Browser Injection from the APM UI.

By default, End User Monitoring will be set to OFF, and the browser injection type set to Correlation. You can change this to other injection types. You can change the injection type of an application server, only if the APM Agent installed is version 1.21 or higher.
To configure the default browser injection type for an application server:

1. From the APM left navigation menu, select **APM Admin** and select **Browser Agent**.
2. In the **Configure End User Monitoring Injection Type Property** section, all the application servers are listed. Optionally, click the Filter icon to search for the required application server.
3. Select the application server to modify, and click the Edit icon in the **Injection Type** column, and select the required injection type.
   - **Reference:** This injects the both the javascript and the correlation cookie automatically, using a reference to a central javascript library.
   - **Correlation:** This injects only the correlation cookie. You can manually inject the javascript to the pages you want to monitor.
   - **None:** No automatic injection is done.
   - **Full:** This injects the full reporting javascript and the correlation cookie, and is not a recommended option.

Configure Attributes for End User Monitoring

You can configure Application Performance Monitoring to report on additional attributes, to enable better classification of different end user activities. The values for these attributes will be reported in various pages of APM, and provide a closer perspective of application performance on the user’s site.

The newly added attributes are displayed on reports on Page list, Page details, Ajax Call list, Ajax Call details, and Session Detail pages. You can configure how the attributes should be populated. These functions will be executed when an Ajax Call or a click is detected. For best results, use attributes to report more details of the application area or application state where an action occurred.
To add attributes to report on:

- In the pages you want to be monitored, include the following Javascript code:

  ```
  apmeum.udfAttribute1 = function()
  {
      // return desired value of attribute 1
      return "Attribute value 1";
  };
  apmeum.udfAttribute2 = function()
  {
      // return desired value of attribute 2
      return "Attribute values 2";
  };
  ```

  These attributes will be included in the reports generated on the monitored pages. By default, `attribute1` will be populated with the top level directory of the URL and `attribute2` will remain empty.
  Example: In the URL `https://www.sample.com/calendar/render`, `attribute1` would be populated with the string *calendar*.

  By default, the document title of the application page is captured as page title, but can be redefined by setting the `apmeum.udfAttributeDt` attribute. In context reporting, the data in `attribute1` is most generic, `udfAttributeDt` is more specific (page level) and `apmeum.udfAttribute2` is assumed to point / describe roughly the area on a screen that is active during an AJAX call or the area that is clicked.

  Example:

  ```
  apmeum.udfAttributeDt = function()
  {
      // return desired value of page title
      return "New Page Title";
  };
  ```

Monitor End User Performance with Manual Browser Injection

You can monitor the experience of your user with your application without the use of an APM Agent.

Browser injection happens automatically if the APM Agent is configured accordingly, but can also be done manually. Manual injection should be used in the following scenarios:

- When there is no APM Agent available for your application server or when you want to monitor static pages. In this case, manual injection will still allow you to see and alert based on page performance. You can see user sessions, and all instances of pages being visited.
- Where automatic injection as performed by the APM Agent does not provide desirable results (for example, when you want a different injection location). In that case, configure the APM Agent for correlation mode. Along with manual injection, the complete functionality available with APM will be available.

If you configure your application to perform manual browser injection and an APM Agent is already present, then you can use one of the following options:

1. Configure the Agent to perform correlation by setting `oracle.apmaas.agent.browser.injectionType=correlation`. This is the preferred option.
2. Use no injection by setting `oracle.apmaas.agent.browser.injectionType=none`. However, this could result in navigation limitations.

The following sections describe how to configure and use manual injection.

- Obtain the Collector URL
- Check Browser Agent Rewrite Rules for Manual Injection
- Manual Injection of APM Javascript Page Tag
- Parameters for Automatic Browser Injection

Obtain the Collector URL

Before you can manually inject the Javascript, you need to obtain the Collector URL that will be used as part of the static script that will be injected.

1. In the left navigation pane, click **APM Admin** and select **Browser Agent**.
2. The Collector URL is listed in the Browser Agent screen.

To configure APM to use a different `collectorUrl` and to transfer data through reverse proxy, see Transferring EUM Data to OMC Collectors.

Check Browser Agent Rewrite Rules for Manual Injection

To check the default value for the Browser Agent Rewrite Rules, do the following:

1. In the left navigation pane, click **APM Admin** and select **Browser Agent**.
2. Scroll down to the Browser Agent Rewrite Rules section from the **Browser Agent** page to view the default value for the Browser Agent Rewrite Rules.

Manual Injection of APM Javascript Page Tag

In some environments it is not possible to use the automatic Javascript injection functionality to monitor end user performance of web pages. Static Javascript monitoring is useful for environments like Siebel, CDN, or when no agent is available for your application server platform.

In may application frameworks or content managements systems it will be sufficient to change a few page templates to change all application pages. In such cases, you can manually inject the APM Javascript page tag into your HTML to monitor end user page performance, without having the APM Agent installed.

When you use this option the page data cannot be correlated to specific server request data. Correlation to server requests is possible only if the agent is configured for *correlation* mode.

> **Note:**
>
> The details of where and how to manually inject the Javascript page tag are specific to the application, OS Platform and Application Server platform.

Insert the static Javascript manually in a HTML page:

1. In the left navigation pane, click Administration and select Browser Agent.
2. Copy the **Javascript for agentless End User Monitoring** provided in the right pane.
3. Open the HTML page you want to set End User Monitoring for, in Page Source mode.
4. Insert the copied script after the closing `</head>` tag. Ensure that you do not add the code within another `<script>` tag.

Insert the static Javascript manually in a PHP page:

If the web page to which you want to manually add the Javascript is a PHP page without HTML tags, add the EUM Javascript at the end of the PHP source code. Here is an example:

```
<?php
phpinfo();
?>

<script>
window.apmeum || (apmeum = {});
apmeum.customerId='emaastesttenant1';
apmeum.baseCollectorUrl='https://abc1.example.com:4443/APMaaSCollector/
external/collector?';
</script>
<script async
crossorigin="anonymous" src="https://abc1.example.com:4443/
APMaaSCollector/external/collector/staticlib/apmeum.js"></script>
```

To optimize performance for users of the monitored application an alternative CDN-backed location of the Javascript library can be specified as follows:

```
<script async crossorigin="anonymous" src="https://
static.oracle.com/cdn/apm/1.47.0/apmeum.js"></script>
```

Parameters for Automatic Browser Injection

Here are the fields that are configured for browser injection during automatic injection:

Area	Sample Initialization	Default Value	Description
Collector root	`oracle.apmaas.agent.collectorRoot=https://customer.itom2.management.pp1.oraclecloud.com/`	Based on values received during provisioning of the agent.	This is the basic collector url that is suffixed with `APMaaSCollector/external/collector` to get the **collector url**, and with `APMaaSCollector/external/collector/staticlib/apmeum.js` to get **jsLibraryPath**.
Collector URL	`oracle.apmaas.agent.collectorUrl=https://customer.itom2.management.pp1.oraclecloud.com/APMaaSCollector/external/collector?`	Based on `collectorRoot` (appending that with `APMaaSCollector/external/collector`)	The default location to send observations to. If this is not set, it is initialized relative to collector Root. Override this value only when using reverse-proxy.
Javascript Library URL	`oracle.apmaas.agent.jsLibraryUrl=https://customer.itom2.management.pp1.oraclecloud.com/APMaaSCollector/external/collector/staticlib/apmeum.js`	Relative to collector URL. Replace the **?** with `/staticlib/apmeum.js`	The injected reference `.js` file. This option should be used if you want to host the APM Javascript library locally for any reason.
Influence observations	`oracle.apmaas.agent.browser.setting="*(click:off),firefox(ajax:off,click:off)"`	`*:on`	The APM Agent monitors observations that are enabled for the site. This can be configured globally or per browser.
Disable injection completely	`oracle.apmaas.agent.enableBrowserAgent=true`	`true`	The goal is to continue monitoring data from the server-side only. If this option is combined with manual injected Javascript, the only missing part is the relation/correlation between browser side and server side monitoring.

Area	Sample Initialization	Default Value	Description
Type of injection	`oracle.apmaas.agent.browser.injectionType="full"`	`reference`	This allows customers to change the way the agent injects data. The available values are: • `full` for agents that allow full injection of Javascript • `reference` used to indicate injection with reference to the central library • `correlation` used to indicate injection using correlation information like adding operation ID into Response Cookie header.
Customer tenant ID	`oracle.apmaas.agent.tenant=fixedtenant`		The tenant ID to use for reporting

Enable End User Monitoring in Siebel Environment

You can enable end user monitoring in a Siebel environment through manual instrumentation.

To enable end user monitoring in a Siebel environment:

1. In the Siebel installation folder, navigate to `ses/siebsrvr/webtempl/ouiwebtempl`.
2. Open the HTML source of the login page, `SWELogin.swt` and insert the APM Javascript manually to the page.
3. Open the HTML source of the container page, `ccpagecontainer.swt` and insert the APM Javascript manually to the page.
4. Configure the Manifest Administration to enable manual injection of the APM Javascript in sections such as **View** and **Applet**.
   a. Create a file, for example, `getSiebelViewName.js`.
   b. Populate the file with the following code:

```
if(typeof (SiebelAppFacade.APMTracker) == "undefined") {
    Namespace('SiebelAppFacade.APMTracker');
    (function(){
      SiebelApp.EventManager.addListner( "postload",
           TrackAPMOnViewLoad, this );
      function TrackAPMOnViewLoad( ){
        try{
          window.apmeum || (apmeum = {});
          apmeum.eventAttributes.getPageName = function() {
              if( SiebelApp.S_App &&
SiebelApp.S_App.GetActiveView() ){
                 return
SiebelApp.S_App.GetActiveView().GetName();
              } else {
                 return "Unable to determine active view";
              }
          }
          apmeum.eventAttributes.getAttr1 = function() {
              if( SiebelApp.S_App &&
SiebelApp.S_App.GetActiveBusObj() ){
                 return
SiebelApp.S_App.GetActiveBusObj().GetName();
               } else {
                  return "Unable to determine active Business
Object";
               }
          }
           // Use 'full name' to identify user
          var un = SiebelApp.S_App.GetProfileAttr("Full Name");
          // DEBUG:     console.log('Tracking info is:'+un);
          if (! window.apmeum.hasOwnProperty("username")) {
                                window.apmeum.username = un;
          }
        }
        catch(error)
        {
           // No-Op
           //  console.log("OOPS check ApmEumTrackUserId logic:
"+error)
       }
      }
    }());
}
```

c. Using the Siebel console, add the file to the Manifest Administration View in the folder `Application/Common/Platform Independent`.

   See Siebel documentation for how to add the file you created to the Siebel environment.

d. Restart all the Siebel servers.

   i. Navigate to `ses/gtwysrvr/bin` and run these commands:

      - Stop the server : `./stop_ns`
      - Start the server : `./start_ns`

   ii. Navigate to `ses/siebsrvr/bin` and run these commands:

      - Stop the server : `./stop_server all`
      - Start the server : `./start_server all`

   iii. Navigate to `web/Oracle_WT1/instances/instance1/bin` and run these commands:

      - Stop the server : `./opmnctl stopall`

- Start the server : `./opmnctl startall`

You can now monitor the required pages in the Siebel environment with the help of the injected APM Javascript.

Configure User Name Reporting

Application Performance Monitoring can provide reports on user names.

Application Administrators can configure Application Performance Monitoring to create reports on user names by following these steps:

1. Identify the original source of the `username`. This depends on the monitored application and authentication mechanism used.
2. Embed scripting in the application pages (or the application landing page). Assign the user name value to the `apmeum.username` variable. This makes the user name available for reporting.

For user names to be reported by APM they have to be made known to APM by setting the `apmeum.username`variable. The original source of the username depends on the monitored application and authentication mechanism used. Below are a few examples of how `username` can be collected in different types of environments:

1. **Get the username from Windows in Internet Explorer:**

   ```
   var WinNetwork = new ActiveXObject("WScript.Network");
   window.apmeum || (apmeum = {} );
   apmeum.username =  WinNetwork.UserName
   ```

2. **Get the username logic from PHP code:**

   ```
   window.apmeum || (apmeum = {} );
   apmeum.username = '<?php echo $username; ?>';
   ```

3. **Get the username from the page:**

   You can use this if the page is using DOM, and the page contains something like `<div id="welcomeMsg">Welcome <user> (last visit <mm-dd-yyyy>)</div>`.

   ```
   var Loginname = document.getElementById("welcomeMsg").innerHTML ;
   var end = Loginname.indexOf("(");
   var nameOnly = Loginname.substring(8, end);
   window.apmeum || (apmeum = {} );
   apmeum.username = nameOnly;
   ```

4. **Get the username from EBS username:**

   You can extract username from your Oracle E-Business Suite (EBS) site.

   a. Log in to the EBS site, and navigate to a page where your log-in name is visible.
   b. Using an editor, view the source of this page. It is recommended that you use an editor that shows the source code without formatting or rendering.
   c. Search for your username in the code. Here are some examples of how you may see the username:

Pattern 1:

```
[...]
<td width="100%">
        <h1 class="x1f" type="OraHeader">
   Welcome <USER NAME>, <DATE>
  </h1>
</td>
[...]
```

Pattern 2:

```
[...]
<td class="x8g">
        <span class="x2v">Logged In As </span>
        <span class="x2u"><USER NAME> </span>
 </td>
[...]
```

d. For the 2 patterns mentioned above, you can use the Javascript code shown below to extract user names. Add this snippet just before the end of the <body> section.

```
<head>
...
...
...
</head>
<body>
...
...
...
<script type='text/javascript' charset='UTF-8'>
  var namefromCookie = apmeum.util.getCookie('EBSUSERNAME');
  if (namefromCookie != null && namefromCookie.length > 0) {
    apmeum.username = namefromCookie;
  } else {
    var spanList1 = document.getElementsByClassName("x1f");
    var spanList2 = document.getElementsByClassName("x2u");
    if (spanList1 != null && spanList1.length > 0) {
      var loginName = spanList1[0].innerHTML;
      apmeum.username = loginName.replace('welcome ','');
    } else if (spanList2 != null && spanList2.length > 0) {
      var loginName = spanList2[0].innerHTML;
      apmeum.username = loginName;
    }
  }
</script>
</body>
```

Tip: Inspect the HTML page's session cookies — In Firefox, you can see this in the *Tools* menu, *Web Developer* option, *Storage Inspector*. Verify that a cookie containing the username is present. If the username is different from the EBSUSERNAME, use the observed name.

Transfer EUM Data to OMC Collectors

APM supports different deployment scenarios for transfer of EUM data to OMC. Listed here are different deployment scenarios and related configuration options.

Determine configuration scenario for your deployment

Scenario 1: If your company allows full internet access to all users, EUM data will be sent to OMC without any special configuration, without a proxy setup.

Scenario 2: If your company limits internet access, but would still like complete EUM data to be sent to OMC, then you should configure your existing firewall or proxy to permit requests to be sent to OMC. See Transferring EUM Data through a Proxy.

Scenario 3: If there are special security or administrative requirements, you can configure a reverse proxy or proxy in front of existing proxies or firewalls. See Transferring EUM Data through a Reverse Proxy.

Transfer EUM Data through a Proxy

In cases where internal users of an enterprise web application have access to that application, but not to the internet in general, proxy rules should be adapted to allow access to the APM collector for all users. In that way, you can send EUM performance data to the APM collector even if users do not have access to the collector otherwise.

You can see the details of the APM collector endpoint to be configured in the APM UI by selecting **APM Admin** and then **Browser Agent**. Enable access to the collectorURL. Configuration of a proxy is vendor specific, see the documentation provided by the proxy vendor for configuration details.

Transfer EUM Data through a Reverse Proxy

The reverse proxy technique allows browsers to find a way through the customer firewall that is open for access only to the OMC Collector. The browser will find the reverse proxy and interact with it like the Oracle cloud end point. You can configure reverse proxy by following these steps:

- Configure APM Agent to use a different collectorUrl
- Configuring a Reverse Proxy
- Collecting Internal IP Addresses

Configure APM Agent to use a different collectorUrl

The APM Agent should override the end point it receives from Oracle cloud with the local end point of the reverse proxy. This can be done by configuring the property for `collectorUrl` in `AgentStartup.properties` file. This setting will override the default value that is retrieved from Oracle cloud through service registry. When there are no `https` sites that require monitoring, set the `collectorUrl` to `http` instead of `https`.

Here is an example of the setting:

```
oracle.apmaas.agent.collectorUrl=https://myproxy.example.com:4443/
APMaaSCollector/external/collector
```

In the example above, replace `myproxy.example.com` and the port number **4443** with the hostname and the port number of the machine where the reverse proxy is installed. Once the `collectorUrl` is updated, bounce the application server(s).

Configure a Reverse Proxy

You can configure a reverse proxy using any reverse proxy configuration tools like NGINX, Squid or WebLogic ProxyPlugin.

> **Note:**
>
> If you have an Oracle HTTP Server setup, skip steps 1-6. If you are using a trusted certificate, skip step 7.

1. Navigate to http://www.oracle.com/technetwork/middleware/webtier/downloads/index.html
2. Accept the license agreement.
3. Scroll to **Oracle WebTier 12cR2** and **Oracle HTTP Server 12.2.1.1**.
4. Select **Linux-64 bit** and download the installer.
5. Follow the documentation to install Oracle HTTP Server at https://docs.oracle.com/middleware/1213/index.html.

   > **Note:**
   >
   > The above document assumes that OHS is installed at `/Oracle`.

6. Start the `nodemanager`.

   ```
   # export DOMAIN_HOME=<WLS Domain home>
   # cd $DOMAIN_HOME/bin
   # nohup ./startNodeManager.sh > nm.out&
   ```

   Example:

   ```
   # export DOMAIN_HOME="/Oracle/Middleware/Oracle_Home/user_projects/
   domains/base_domain"
   # cd $DOMAIN_HOME/bin
   # nohup ./startNodeManager.sh > nm.out&
   ```

7. Add the HTTP Server certificate to wallet.
   a. Prepare the wallet:

      ```
      # export DOMAIN_HOME=<WLS Domain home>
      # cd $DOMAIN_HOME/config/fmwconfig/components/OHS/instances/ohs1/
      keystores
      # mkdir proxy
      # $DOMAIN_HOME/../../../oracle_common/bin/orapki wallet create -
      wallet . -auto_login_only
      ```

Example:

```
# export DOMAIN_HOME=/Oracle/Middleware/Oracle_Home/
user_projects/domains/base_domain
# cd $DOMAIN_HOME/config/fmwconfig/components/OHS/instances/ohs1/
keystores
# mkdir proxy
# $DOMAIN_HOME/../../../oracle_common/bin/orapki wallet create -
wallet . -auto_login_only
```

b. Get the Certificates and add to the wallet:

   Method 1:

```
# echo -n | openssl s_client -connect <OMC collector URL>:<port>
| sed -ne '/-BEGIN CERTIFICATE-/,/-END CERTIFICATE-/p' > /tmp/
proxy_certificate.cert
```

   Method 2:

   i. Install (or update) the CA certificates:

```
# yum update ca-certificates
```

   ii. Split the bundle CA root file containing all certificates into separate files. (`orapki` cannot handle a bundle.)

```
# awk 'BEGIN {c=0;} /BEGIN CERT/{c++} { print > "cert." c
".pem"}' < /etc/pki/tls/certs/ca-bundle.crt
```

   iii. Load the individual certificates into the OHS proxy wallet.

```
# for i in `ls cert.*.pem`; do / Oracle/Middleware/
Oracle_Home/oracle_common/bin/orapki wallet add -wallet . -
cert $i -trusted_cert -auto_login_only; done
```

8. Retrieve the APM `collectorUrl`.

   In the left navigation pane on the APM UI, click **Administration** and select **Browser Agent**. The `collectorUrl` is displayed in the Browser Agent screen. .

9. Configure HTTPS reverse proxy with Oracle HTTP Server.

   a. In an editor, open the `ssl.conf` file located in the folder `ohs1`.

```
# export DOMAIN_HOME=<WLS Domain home>
DOMAIN_HOME/config/fmwconfig/components/OHS/instances/ohs1
```

   Example:

```
# export DOMAIN_HOME="/Oracle/Middleware/Oracle_Home/
user_projects/domains/base_domain"
# vi $DOMAIN_HOME/config/fmwconfig/components/OHS/instances/ohs1/
ssl.conf
```

b. Configure the SSL reverse proxy in an existing virtualhost definition section.

   Example:

```
...
<VirtualHost *:4443>
  <IfModule ossl_module>
   #  SSL Engine Switch:
   #  Enable/Disable SSL for this virtual host.
   SSLEngine on
   SSLProxyEngine on

SSLProxyWallet "${ORACLE_INSTANCE}/config/fmwconfig/components/$
{COMPONENT_TYPE}/instances/${COMPONENT_NAME}/keystores/proxy"
   SSLProxyVerify none
   ProxyPass "/APMaaSCollector" "https://cloud_APM_Collector/
APMaaSCollector"
   ProxyPassReverse "/APMaaSCollector" "https://
cloud_APM_Collector/APMaaSCollector"
   ProxyPreserveHost On
   ProxyRequests off
   ...
</VirtualHost>
...
```

c. Replace the URL `https://cloud_APM_Collector/APMaaSCollector` with the actual collector URL on Oracle cloud from step 8.

   > **Note:**
   >
   > To use HTTP communication between browser and Reverse Proxy, comment out the `SSLEngine on` line.

d. Start `ohs1` component.

```
# export DOMAIN_HOME=<WLS Domain home>
# cd DOMAIN_HOME/bin
# ./startComponent.sh ohs1
```

   Example:

```
# export DOMAIN_HOME=/Oracle/Middleware/Oracle_Home/
user_projects/domains/base_domain
# cd DOMAIN_HOME/bin
# ./startComponent.sh ohs1
```

Collect Internal IP Addresses

In internal company networks that are monitored with EUM, you can view the client IP addresses by making these configurations.

1. Configure your proxy or gateway device to add the `X-ORACLE-CLIENT-IP` header.
2. Populate this header with the real IP address of the client.

Configuration details depend on the device or software being used.

> **Note:**
>
> Reporting on geographic locations is currently not supported for internal IP addresses.

Troubleshoot End User Monitoring

If you run into problems while using End User Monitoring (EUM), here are some tips to troubleshoot the issues.

EUM Data not Appearing in the APM UI

If the EUM data is not appearing in the APM UI, follow these steps to ensure that the data is getting collected.

1. Check Injection Type.
   - Check if you have enabled Javascript injection for the APM Agent. Starting with Oracle Management Cloud 1.21, the default injection type is **Correlation** — which means that Javascript does not get injected automatically to every page, but agents have to be actively configured to perform Javascript injection. To change the injection type, see Enable and Configure End User Monitoring.
   - Alternatively, ensure that the Javascript is manually added to the required application pages. See Monitor End User Performance with Manual Browser Injection.

     Select **Reference** as the **Injection Type**.
2. Check if Javascript is injected.
   - After checking the injection type as listed above, check if the Javascript is getting injected into the application pages.
   - If the EUM data is still not appearing, check the HTML source of the application page that should be monitored, to validate that the Javascript is present. In Firefox and Chrome, right click on the page, and select **View Page Source.** In Internet Explorer, right click and select **View Source**.
   - In the HTML source, search for the string `baseCollectorUrl`. If you cannot locate the string, then the Javascript is not correctly injected. This could be because of any of these factors:
     - EUM JavaScript will be injected by APM agents only if the **Injection Type** is configured for *Full* or *Reference* mode. For further information see, Enabling and Configuring End User Monitoring. If you prefer manual instrumentation of application pages, then leave **Injection Type** configured as *Correlation*. See Monitoring End User Performance through Manual Browser Injection for details.
     - If Javascript does not appear in the application pages once APM agents are configured as described above, then verify that the APM agents are deployed to all application servers delivering application pages.

- For static HTML pages, make sure that the pages cached via content delivery networks (CDN) or content management systems are refreshed.
- If the Javascript was manually injected, ensure that all the application pages are properly instrumented, by ensuring that the Javascript is identical with the Javascript snippet shown in the Oracle Application Performance Monitoring UI. (In the UI, click **APM Admin**, and select **Browser Agent**.)

```
<script>
   window.apmeum || (apmeum = {});
   apmeum.customerId='<tenant id>';
   apmeum.baseCollectorUrl='https://<tenant id>.<data center 
specific>.oraclecloud.com/APMaaSCollector/external/collector?';
   </script>
   <script async src="https://<tenant 
id>.<data center specific>.oraclecloud.com/APMaaSCollector/
external/collector/staticlib/apmeum.js"></script>
```

- If Javascript is correctly injected, but the EUM data is still not visible, then proceed to check for issues with collector availability, as described below.
- If manual injection is used for an application where the back-end is monitored by APM Agents, and there are no server requests shown for pages and Ajax calls, then check if agents are configured for injection type *correlation*. This is required for pages and Ajax calls to be associated with server requests.

3. Is the collector Javascript downloaded and executed?

- By default, the Javascript will be downloaded from the collector location `https://<tenant id>.<data center specific>.oraclecloud.com/APMaaSCollector/external/collector/staticlib/apmeum.js`.

  Some sites have additional protection against remote hosted Javascript to prevent cross-site-scripting. In this scenario, copy the script and host it locally. Configure the `oracle.apmaas.agent.jsLibraryUrl` setting in the `apmagent` properties file.

  Edit the `apmagent` properties file, and add the location of the locally hosted `apmeum.js` file:

```
oracle.apmaas.agent.jsLibraryUrl=https://<www.example.com>/
apmStaticJS/apmeum.js))
```

- Usually, browsers report blocked Javascript executions in the web console as a warning. Check the web console to spot the issue. See more information on the internet about Content Security Policy. For example you can search for Content Security Policy (CSP) in Mozilla Developer Network.

4. Is the collector reachable?

- Ensure that the EUM events are being sent to the `ApmClientCollector`. This is a pre-requisite for the EUM data to be displayed in the APM UI. You can check this by starting the Developer Console.
- In most browsers, you can start the Developer Console by pressing the F12 key, and selecting the **Network** tab.

- In the **Network** tab, load or reload one of the application pages to be monitored by APM. The requests shown should now contain requests to the collector.

  Example: `https://<tenant>.<data center specific>.oraclecloud.com/APMaaSCollector/external/collector`

- If you don't see a 200 or 204 HTTP response code for requests to the collector, then that browser isn't able to access the collector. This could be caused by network configuration that limits internet access. When the internet access for users is limited, a reverse proxy may have to be set up inside the network to allow EUM events to be sent to Oracle Management Cloud. See Transferring EUM Data to OMC Collectors.

- In some cases, the CSP policy blocks interaction with collector, and is usually reported in the web console as warnings.

5. Static content is not monitored by default.

   Server requests for pages that are considered as static content are not monitored by default. File types considered as static content are the following: `".bmp", ".css", ".png", ".swz", ".jar", ".htm", ".html", ".dtd", ".mpeg", ".jpg", ".dat", ".mpg", ".mid", ".properties", ".js", ".ico", ".class", ".tif", ".gif", ".jpeg", ".swf", ".cur", ".woff".`

   To change this configuration, see Monitoring a Web Application through Servlet Monitoring.

Browser Agent Interfering with the Monitored Application

Sometimes, enabling EUM can interfere with the smooth functioning of your application.

If Javascript is available in HTML pages as described above, and EUM metrics are visible in the APM UI, but instrumentation results in issues with functionality of the monitored application, try these tips:

- For manual injection, check where the Javascript is placed. The script should be placed directly after the closing `</head>`tag or at the very end of the page. Ensure that it is not placed within another `<script>` tag.
- When the Javascript is interfering with specific pages only, then turn off the Javascript injection for affected pages.

  To disable Javascript injection for a specific URL pattern (Example: `/test/mysample`):

  1. Edit `apmagent/conf/BrowserAgent.json`
  2. Add the following section. The change will be effective after about 30 seconds, without a restart of the application server.

     ```
     "urlConfigs": [
         {
             "patternString" : "/test/mysample",
             "config" : {
                 "extensionGroup" : "default",
                 "patternMatching" : [{
                     "patternString" : "/test/mysample",
                     "excluded" : true,
                     "includeQueryString": false
     ```

```
                    }]
                }
            }
        ]
```

In some cases the interaction between the APM Javascript and the application's Javascript causes problems. On Internet Explorer, you can resolve this by disabling the tracking of particular events from the browser.

Edit the `apmagent` properties file by adding the browser setting property:

```
oracle.apmaas.agent.browser.setting=ie(ajax:off, click:off)
```

Validating Customizations

You can customize the data that your application sends to Oracle Management Cloud. Validate the customization you've done through these steps:

- Validating `apmeum` object
- Verify `username` property across multiple browsers
- Validating information sent to Oracle Management Cloud

Validating `apmeum` object:

After you have added the customization code, the `apmeum` object will be updated. Use the developer tools of your browser to validate the contents of the `username` property. The below verification is applicable to most of the browsers.

Verify `username` property across multiple browsers:

To validate the `username` property, in the developer tools window of your browser, enter:

```
window.apmeum.username
```

If the customization code has not been updated correctly, you will see the following message:

```
window.apmeum.username
undefined
```

If the customization code has been updated correctly, the user name is displayed as below:

```
window.apmeum.username
<username>
```

Validating information sent to Oracle Management Cloud

The information from `apmeum` object and the metrics are combined into messages that are sent to the `apm collector` in Oracle Management Cloud. This is visible in browsers by requests that are made to `/APMaaSCollector/external/collector` URLs.

You can view this information in the *Network* tab of the *Developer Tools* in most browsers. To validate, check the request body to ensure that the specific test is listed. You can also filter the URLs to view only the relevant ones.

Data in End User Monitoring Reports

Application Performance Monitoring collects different data from end user monitoring and presents it in reports.

Through End User Monitoring, there are three categories of dimensions available for reporting - for sessions, pages and Ajax calls.

Table 14-1 Data dimension in End User Monitoring

APM Object, Category	Dimension
Pages > Geographic information derived from IP addresses	• Continent • Country • Region • City (City information is exposed in session diagnostics only) • ISP (ISP information is exposed in session diagnostics only) • IP (IP information is exposed in session diagnostics only)
Pages > Client information derived from user-agent HTTP header	• Browser type (Firefox, IE, etc) • Browser version • Device type (desktop, mobile) • OS (Windows, MacOS, etc.) • OS Version • Screen size
Pages > Application related dimensions	• URL (with URL parameters and ID values removed) • Domain name • Page title • Attribute 1 By default, it is the top level URL directory. • Attribute 2 By default,empty. Attribute 1 and attribute 2 can be populated via JavaScript of the applications.
Sessions	Additional attribute `user name` that can be populated with additional JavaScript instrumentation. For details, see Configure User Name Reporting.
Ajax Calls	The HTTP status code is available as an additional reporting dimension.

Table 14-2 Metrics available in End User Monitoring data

APM Object and Category	Dimension
Pages > Counts	• Number of page views • Number of 'frustrating' page views • Number of 'tolerable' page views • Number of 'good' page views For 3 counts and above, page views are categorized based on configurable response time thresholds. • Number of page clicks • Number of Ajax calls • Number of Ajax call errors • Number of JavaScript errors
Pages > Metrics	• Load time (Based on W3C Navigation Timing, `loadEventEnd.`) • Interactive time (Based on W3C Navigation Timing, `domInteractive` - indicates when page becomes usable.) • First Byte time (Based on W3C Navigation Timing, `responseStart`. For aggregated data, min and max values available for 3 metrics and above. • Viewing time
Ajax Calls > Counts	• Number of Ajax Requests • Number of 'frustrating' Ajax calls • Number of 'tolerable' Ajax calls • Number of 'good' Ajax calls For the 3 counts and above page views are categorized based on configurable response time thresholds. • Number of related page views • Number of related page clicks • Number of Ajax errors
Ajax Calls > Metrics	• Load Time • Fetch Time

15 Configure Data Collection and Privacy Controls

Application Performance Monitoring allows you to configure privacy settings and control data collection.

This chapter talks about the tools within Application Performance Monitoring that enable you to comply with local regulations with respect to privacy, data collection and processing and storage of data. Using these tools you can control how much of your users' personal data is collected, stored and viewed, thereby complying to the applicable legal requirements.

- Configure Do Not Track Settings
- Configure Privacy Settings

Configure Do Not Track Settings

Administrators can provide an option to users of Application Performance Monitoring to disable tracking.

Users of an application are tracked in order to provide useful and reliable reporting. But users may want to opt out of tracking for reasons of privacy or due to regulatory requirements. Administrators can provide a choice, and enable users to choose their privacy settings.

Administrators can add applicable business logic in their application so that the value of the parameter `window.apmeum.obs` can be determined as per the user's preference. Through this parameter, APM allows the user to choose to be or not to be tracked. If the user chooses not to be tracked, the value of the parameter `window.apmeum.obs` will be set to 0.

The following is an example of how the parameter `window.apmeum.obs` can be used in the `apmeum.js` file.

```
<html>
    <head>

            // function to get cookie
            function getCookie(cname) {
                var name = cname + "=";
                var decodedCookie = decodeURIComponent(document.cookie);
                var ca = decodedCookie.split(';');
                for(var i = 0; i <ca.length; i++) {
                    var c = ca[i];
                    while (c.charAt(0) == ' ') {
                        c = c.substring(1);
                    }
                    if (c.indexOf(name) == 0) {
                        return c.substring(name.length, c.length);
                    }
                }
                return "";
            }

            //this method is for recording user's preference of being 
tracked
            function setObsTriggered(doNotTrack, exdays/*expires day*/){
                if(doNotTrack != undefined){
                    if(doNotTrack == true){
                        window.apmeum = window.apmeum || {};
                        apmeum.obs = 0;
                    }
                    //calculate expires date
                    var d = new Date();
                    d.setTime(d.getTime() + (exdays*24*60*60*1000));
                    var expires = "expires="+ d.toUTCString();

                    //set path=/ so that all pages under a web project 
can access the doNotTrack cookie if avaiable
                    document.cookie = "doNotTrack=" + doNotTrack + ";" 
+ expires +  "; path=/;"
                }

            }

            function doNotTrackPrompt(){
                var r = confirm("Can I track your behaviour data for 
helpping improve user experience?");
                if (r == true) {

setObsTriggered(true,DEFAULT_EXPIRE_DAY_FOR_DO_NOT_TRACK);
                } else {

setObsTriggered(false,DEFAULT_EXPIRE_DAY_FOR_DO_NOT_TRACK);
                }
            }


            function isDoNotTrackSet(){
                 //check if cookie doNotTrack is available to adjust 
observation state(i.e. apmeum.obs) for current page
                var doNotTrack = getCookie("doNotTrack");

                if(doNotTrack != undefined && doNotTrack != ''){
                    if(doNotTrack == 'true'){
                        window.apmeum = window.apmeum || {};
                        apmeum.obs = 0;
                    }
                    return true;
                }
                else return false;
            }

            $(document).ready(function(){
                if(!isDoNotTrackSet()) //if doNotTrack is not set yet, 
a prompt will be popped up.
                    doNotTrackPrompt();
            });

        </script>

    </head>

    <body>
        <!--
            Client's page.
        -->
    </body>
</html>
```

In the above scenario, if the main page of a website records a *Do Not Track*, the same value is carried forward to its child pages too. If a user chooses *Do Not Track* for `www.samplepage.com`, the same preference would be applied for `www.samplepage.com/cart`.

Configure Privacy Settings

Administrators can provide an option to users of Application Performance Monitoring to configure privacy preferences.

You can configure privacy settings in Application Performance Monitoring to comply with legal requirements.

1. From the APM left navigation menu, select **APM Admin** and select **Privacy Settings** .
2. Select the privacy option as per your requirement:

Privacy Setting	Description
Discard IP addresses, obtain geo-location data	This is the default selection. Select this option to mask users’ IP addresses and also discard the IP address data. APM UI will display the IP address as —.
Retain IP addresses and obtain geo-location data	Select this option to retain users’ IP addresses.
Discard IP addresses and do not obtain geo-location data	Select this option to discard both IP address and geo-location data of all the users.

3. Personal Identifiable Information: To avoid storing any personally identifiable information (PII), you can select if you want to store private information such as full URLs, page titles and click names as per your requirement. Note that this does not affect Web Application data.

Privacy Setting	Description
Store full URLs, not only the domain	Check this option to store full URLs.

	If you check this option, APM UI will display `https://<domain name>:<port>/<Directory Path...>/<File Path>` in Page, Ajax and Session pages. If you don't check this option, APM UI will display only `https://<domain name>:<port>` in Page, Ajax and Session pages. By default, this option is unchecked if you did not use APM before.
Store page titles and click names	Check this option to store page titles and click names. If you don't check this option, APM UI won't collect the page titles. By default, this option is unchecked if you did not use APM before.

4. Click **Save**.

16
Upgrade APM Agents

A user with Application Performance Monitoring administrator role can update to the latest version of the Agent.

Topics:

Upgrade APM Java Agent

This section discusses how you can upgrade the APM Java Agent.

Upgrade APM Java Agent

1. Stop the server where the APM Java Agent to be upgraded is installed. Ensure you are logged in as the same user that installed the initial APM Java Agent.
2. Download the agent install software. See the install instructions for the relevant APM Java Agent.
3. Optionally backup the existing APM Java Agent. Note that the `apmagent/config` and `apmagent/lib` folders will be backed up automatically during the upgrade.
4. Install and provision the APM Java Agent for your administration server. Provision the new APM Java Agent to the same destination.

   The provisioning script will search for `domain home/apmagent` (where `domain home` is the directory the user specified in the `-d` parameter.) You will be prompted to Overwrite, or Upgrade existing APM Java Agent.

   a. **o — Overwrite** - New APM Java Agent will be installed over the existing one.
   b. **u — Upgrade** - The APM Java Agent is upgraded to new version, and all customized properties are retained. Upgrade does the following:
      - Old agent `libs` will be backed up: `apmagent/lib` into `apmagent/lib.backup`
      - Old agent `libs` will be backed up: `apmagent/config` into `apmagent/config.backup`
      - New agent files will be extracted into `apmagent,` but `apmagent/config` files will not be replaced.
      - Certain parameters in the `apmagent/config` files will still be updated according to the new installation settings (like server URL, etc.)
   c. **q— Quit** - The script exits without modifying the installed APM Java Agent.
5. Start the application server.

Upgrade APM .Net Agent

This section discusses how you can upgrade APM .Net Agents.

Upgrade APM .Net Agent

Before upgrading the APM .Net Agent, make sure you take a backup of the following system configuration files:

- `C:\Windows\System32\inetsrv\config\applicationHost.config`
- `C:\Windows\Microsoft.NET\Framework\v4.0.30319\Config\web.config`
- `C:\Windows\Microsoft.NET\Framework\v2.0.50727\CONFIG\web.config`
- `C:\Windows\Microsoft.NET\Framework64\v4.0.30319\Config\web.config`
- `C:\Windows\Microsoft.NET\Framework64\v2.0.50727\CONFIG\web.config`

To upgrade the APM .Net Agent:

1. Stop the IIS server.

   ```
   iisreset /stop
   ```

2. Execute the APM .Net Agent installer.
   a. Ensure that the `OMC.ini` file is in the same directory as the `ApmAgent.msi` file.
   b. Run the `ApmAgent.msi` executable. The APM .Net Agent installation wizard guides you through the installation process.
   c. Specify the installation directory. You can use the same directory used for the previous installation, or specify a new one.
   d. Specify the host name.
   e. Click **Install**.
3. Start your IIS server.

   ```
   iisreset /start
   ```

Upgrade APM Node.js Agent

This section discusses how you can upgrade APM Node.js Agent.

Upgrade APM Node.js Agent

Prerequisites:

- Ensure that the environment variables are set appropriately.
- Ensure that none of the files in the `node_modules/oracle-apm` folder are open.
- Stop all the APM Node.js Applications.
- Ensure you are logged in as the same user that installed the initial APM Node.js Agent.

To upgrade APM Node.js Agents:

1. Download the latest APM Node.js Agent software and extract the contents of the ZIP file to a local or shared directory.
2. Install the APM agent install software.

   For information on the above steps 1 and 2, see Install and Configure APM Node.js Agents.
3. Optionally, backup the existing APM Node.js Agent from `$NODE_PATH`.
4. Install and provision the APM Node.js Agent by running the provisioning script.

   APM Node.js Agent displays the list of changes made by the upgrade process, and prompts for confirmation. On confirmation, the APM Node.js Agent is upgraded to the new version, and all customized properties are retained. The `oracle-apm` directory and the `oracle-apm-config.json` file in `$NODE_PATH` are backed up to `${STAGE_DIR}/temp`.

   > **Note:**
   >
   > Review the `hostName` property, since this value will overwrite the `hostName` which was used by the agent prior to upgrade.
   >
   > The upgrade computes a default `hostname` and uses the same. To override this default, rerun the provisioning script with `ORACLE_HOSTNAME` argument.

The following changes occur when you choose to upgrade APM Node.js Agent:

1. A new version of APM Node.js Agent software will be installed in the `node-modules` folder.
2. Updated `oracle-apm-config.json` will be copied over from `STAGE_DIR` to `node-modules/oracle-apm/data` folder. Existing file will be backed up with a `.backup` extension. Custom properties edited previously like `proxy params` etc will be copied over to the new file.
3. All `.cer` certificate files will be converted into the correct format and copied over to `node-modules/oracle-apm/data` folder.
4. Start the Node.js applications. The required `config` files will be copied to the respective `NODE_APP_HOME/oracle-apm/data` folder.

Troubleshooting Upgrade Issues

The upgrade overwrites `oracle-apm` agent files in the `%NODE_PATH%` folder. If any file from this folder (for example, `oracle-apm-config.json` file) is open during the upgrade, the following `npm` error occurs, and the upgrade process stops.

```
npm ERR! code EPERM
npm ERR! errno -4048
npm ERR! syscall rename

npm ERR! Error: EPERM: operation not permitted, rename '%NODE_PATH%
\oracle-apm' -> '%NODE_PATH%\.oracle-apm.DELETE'
```

With the above error, the `oracle-apm` folder is deleted and hence, a rerun of the script is treated as a new install. Any locally modified properties are lost due to the rerun.

Workaround

Follow this workaround before attempting a rerun of the provisioning script:

1. The backup copy of the old `oracle-apm` folder is available in the backup folder - `${STAGE_DIR}/temp` (on Linux) and `%TEMP% folder` (on Windows). Copy this folder into `node_modules.`
2. Ensure that none of the files from the `node_modules/oracle-apm` folder are open, and rerun the provisioning script.

Upgrade APM Ruby Agent

Follow these steps to upgrade the APM Ruby Agent.

Upgrade APM Ruby Agent

1. Download the latest APM Ruby Agent from the **Oracle Management Cloud UI**.
2. Install the new agent gem.

   ```
   gem install oracle_apm-1.xx.x.gem
   ```

3. Verify that it has been added to the gem library:

   ```
   gem list oracle_apm
       oracle_apm (1.yy.y) # other old version(s)
       oracle_apm (1.xx.x) # new version
   ```

4. Edit the application’s Gemfile and update the agent gem version specifier

   ```
   gem 'oracle_apm', '~> 1.xx.x’
   ```

5. Restart your application server.

17
Uninstall APM Agent

A user with Application Performance Monitoring administrator role can uninstall the APM Agent.

Topics:

- Uninstall APM Java Agent
- Uninstall APM .Net Agent
- Uninstall APM Node.js Agent
- Uninstall APM Ruby Agent
- Disable and Remove Agents for Licensing Purposes

Uninstall APM Java Agent

This section discusses how you can uninstall APM Java Agents.

- Remove APM Java Agents from Oracle WebLogic Server
- Remove APM Java Agents from WebSphere
- Remove APM Java Agents from Apache Tomcat
- Remove APM Java Agents from JBoss
- Remove APM Java Agents from Jetty

Remove APM Java Agents from Oracle WebLogic Server

To disable and remove APM Java Agents from a WebLogic domain, complete the following tasks:

1. **Task 1: Disable the APM Java Agent in the WebLogic Server Domain**
   a. Stop the WebLogic Server:

      ```
      % cd $DOMAIN_HOME/bin
      % ./stopWebLogic.sh
      ```

   b. Remove the edited version of the `startWebLogic.sh` script, and replace it with the original script that was backed up before you installed the APM Java Agent:

      ```
      % mv startWebLogic.sh.orig startWebLogic.sh
      ```

   c. Restart the WebLogic Server:

      ```
      % cd $DOMAIN_HOME
      % nohup ./startWebLogic.sh >& startup.log &
      ```

> **Note:**
>
> Note that in the above script you are using the `$DOMAIN_HOME` version of `startWebLogic.sh`, even though you had earlier edited the `$DOMAIN_HOME/bin` version. Invoking this script from one level higher will in fact invoke the script from a lower level.

2. **Task 2: Delete the APM Java Agent Software from the WebLogic Server Domain**
   1. Remove the directory where the APM Java Agent files were extracted:

      ```
      % cd $DOMAIN_HOME
      % rm -r apmagent
      ```

   2. Remove the directory where you initially extracted the APM Java Agent installation software.
3. **Task 3: Remove APM Java Agent References from Oracle Management Cloud**
   1. On the **Oracle Management Cloud Agents** page, click **APM Agents** on the left navigation pane.
   2. On the **APM Agents** page, select the APM Java Agent that you want to remove. Use the **Search** feature to search for a specific APM Java Agent.
   3. On the right side of the page, click the **Actions** menu and select **Remove**.

Remove APM Java Agents from WebSphere Server

To disable and remove APM Java Agents from your WebSphere application server, complete the following tasks:

1. **Task 1: Disable the APM Java Agent in the WebSphere Server**
   1. From your WebSphere Admin console, click the **Servers** tab and select the server on which you want to provision the APM Java Agent.
   2. Expand **Java and Process Management** tab and select **Process Definition**.
   3. Under **Additional Properties** tab, select **Java Virtual Machine**.
   4. In the Generic JVM arguments field, remove the following line:

      ```
      -javaagent:\$WAS_HOME/apmagent/lib/system/
      ApmAgentInstrumentation.jar -Dws.ext.dirs=\$WAS_HOME/apmagent/lib/
      agent/ApmEumFilter.jar
      ```

   5. Remove the edited version of the `server.policy` file, and replace it with the original script that was backed up before you installed the APM Java Agent:

      ```
      % mv server.policy.sh.orig server.policy.sh
      ```

f. Restart the WebSphere Server:

```
% cd $WAS_HOME/bin
./stopServer.sh <servername>
./startServer.sh <servername>
```

> **Note:**
>
> Note that in the above script you are using the `$WAS_HOME` version of the `server.policy` file, even though you had earlier edited the `$WAS_HOME/bin` version. Invoking this script from one level higher will in fact invoke the script from a lower level.

2. **Task 2: Delete the APM Java Agent Software from the WebSphere Server**

   a. Remove the directory where the APM Java Agent files were extracted:

   ```
   % cd $WAS_HOME
   % rm -r apmagent
   ```

   b. Remove the directory where you initially extracted the APM Java Agent installation software.

3. **Task 3: Remove APM Java Agent References from Oracle Management Cloud**

   a. On the **Oracle Management Cloud Agents** page, click **APM Agents** on the left navigation pane.

   b. On the **APM Agents** page, select the APM Java Agent that you want to remove. Use the **Search** feature to search for a specific APM Java Agent.

   c. On the right side of the page, click the **Actions** menu and select **Remove**.

Remove APM Java Agents from Apache Tomcat

To disable and remove APM Java Agents from your Apache Tomcat application server, complete the following tasks:

1. **Task 1: Disable the APM Java Agent in the Apache Tomcat Server**

   a. Stop the Apache Tomcat Server:

   ```
   % cd $CATALINA_HOME/bin
   % ./shutdown.sh
   ```

   b. Remove the edited version of the `catalina.sh` file, and replace it with the original script that was backed up before you installed the APM Java Agent:

   ```
   % mv catalina.sh.orig catalina.sh
   ```

c. Restart the Tomcat Server:

```
% cd $CATALINA_HOME/bin
% ./shutdown.sh
% ./startup.sh
```

> **Note:**
>
> Note that in the above script you are using the `$CATALINA_HOME` version of the `catalina.sh` file, even though you had earlier edited the `$CATALINA_HOME/bin` version. Invoking this script from one level higher will in fact invoke the script from a lower level.

2. **Task 2: Delete the APM Java Agent Software from the Tomcat Server**

   a. Remove the directory where the APM Java Agent files were extracted:

   ```
   % cd $CATALINA_HOME
   % rm -r apmagent
   ```

   b. Remove the directory where you initially extracted the APM Java Agent installation software.

3. **Task 3: Remove APM Java Agent References from Oracle Management Cloud**

   a. On the **Oracle Management Cloud Agents** page, click **APM Agents** on the left navigation pane.

   b. On the **APM Agents** page, select the APM Java Agent that you want to remove. Use the **Search** feature to search for a specific APM Java Agent.

   c. On the right side of the page, click the **Actions** menu and select **Remove**.

Remove APM Java Agents from JBoss

To disable and remove APM Java Agents from your JBoss application server, complete the following tasks:

1. **Task 1: Disable the APM Java Agent in the JBoss Server**

   a. Stop the JBoss Server:

   ```
   % cd $JBOSS_HOME/bin
   % ./jboss-cli.sh -c :shutdown
   ```

   b. Remove the edited version of the `standalone.conf` file, and replace it with the original script that was backed up before you installed the APM Java Agent:

   ```
   % mv standalone.conf.orig standalone.conf
   ```

   c. Restart the JBoss Server:

```
% cd $JBOSS_HOME/bin
% ./jboss-cli.sh -c :shutdown
% nohup ./standalone.sh -b 0.0.0.0&> startup.log &
```

> **Note:**
>
> Note that in the above script you are using the `$JBOSS_HOME` version of the `standalone.conf` file, even though you had earlier edited the `$JBOSS_HOME/bin` version. Invoking this script from one level higher will in fact invoke the script from a lower level.

2. **Task 2: Delete the APM Java Agent Software from the Tomcat Server**
   a. Remove the directory where the APM Java Agent files were extracted:

```
% cd $CATALINA_HOME
% rm -r apmagent
```

   b. Remove the directory where you initially extracted the APM Java Agent installation software.
3. **Task 3: Remove APM Java Agent References from Oracle Management Cloud**
   a. On the **Oracle Management Cloud Agents** page, click **APM Agents** on the left navigation pane.
   b. On the **APM Agents** page, select the APM Java Agent that you want to remove. Use the **Search** feature to search for a specific APM Java Agent.
   c. On the right side of the page, click the **Actions** menu and select **Remove**.

Remove APM Java Agents from Jetty

To disable and remove APM Java Agents from your Jetty application server, complete the following tasks:

1. **Task 1: Disable the APM Java Agent in the Jetty Server**
   a. Stop the Jetty Server.

   Remove the edited version of the `java -jar start.jar` file, and replace it with the original script that was backed up before you installed the APM Java Agent:

```
% mv java -jar start.jar.orig java -jar start.jar
```

   b. Restart the Jetty Server.

> **Note:**
>
> Note that in the above script you are using the `$JETTY_HOME` version of the `java -jar start.jar` file, even though you had earlier edited the `$JETTY_HOME/bin` version. Invoking this script from one level higher will in fact invoke the script from a lower level.

2. **Task 2: Delete the APM Java Agent Software from the Jetty Server**
   a. Remove the directory where the APM Java Agent files were extracted:

   ```
   % cd $JETTY_HOME
   % rm -r apmagent
   ```

   b. Remove the directory where you initially extracted the APM Java Agent installation software.
3. **Task 3: Remove APM Java Agent References from Oracle Management Cloud**
   a. On the **Oracle Management Cloud Agents** page, click **APM Agents** on the left navigation pane.
   b. On the **APM Agents** page, select the APM Java Agent that you want to remove. Use the **Search** feature to search for a specific APM Java Agent.
   c. On the right side of the page, click the **Actions** menu and select **Remove**.

Uninstall APM .Net Agent

This section discusses how you can uninstall APM .Net Agents.

Remove APM .Net Agent

An administrator can remove APM .Net Agents by completing the following tasks:

- Task 1: Remove APM .Net Agent
- Task 2: Remove APM .NET Agent References from Oracle Management Cloud

Task 1: Remove APM .Net Agent

1. Stop the IIS server.

   ```
   iisreset /stop
   ```

2. You can remove APM .Net Agent in any of the following ways:
   - Navigate to the installation directory and run the `ApmAgent.msi` executable. The APM .Net Agent uninstallation wizard guides you through the uninstallation process.

     OR
   - Navigate to the Control Panel and remove **Oracle APM .NET Agent**.

3. For silent uninstallation, run the following command from the installation directory:

   ```
   msiexec /quiet /log uninstall.log /x ApmAgent.msi
   ```

4. If you did not stop the IIS server before you started the uninstallation, the uninstallation wizard displays a message that the `APMAgent\config` folder and the related `.json` files have not been deleted. Delete this folder and the `.json` files manually to remove all the configuration settings. If the folder is not deleted, these configuration settings will be used the next time you install Application Performance Monitoring.

Task 2: Remove APM .NET Agent References from Oracle Management Cloud

1. On the **Oracle Management Cloud Agents** page, click **APM Agents** on the left navigation pane.
2. On the **APM Agents** page, select the APM .Net Agent that you want to remove. Use the **Search** feature to search for a specific APM .Net Agent.
3. On the right side of the page, click the **Actions** menu and select **Remove**.

Uninstall APM Node.js Agent

This section discusses how you can uninstall APM Node.js Agents.

Upgrade APM Node.js Agent

Prerequisites:

- Ensure that the environment variables are set appropriately.
- Ensure that none of the files in the `node_modules/oracle-apm` folder are open.
- Stop all the APM Node.js Applications.
- Ensure you are logged in as the same user that installed the initial APM Node.js Agent.

To upgrade APM Node.js Agents:

1. Download the latest APM Node.js Agent software.
2. Extract the contents of the ZIP file to a local or shared directory.
3. Download the agent install software.

   For information on the above steps 1 to 3, see Installing and Configuring APM Node.js Agents.

4. Optionally, backup the existing APM Node.js Agent from `$NODE_PATH`.
5. Install and provision the APM Node.js Agent by running the provisioning script.

   APM Node.js Agent displays the list of changes made by the upgrade process, and prompts for confirmation. On confirmation, the APM Node.js Agent is upgraded to the new version, and all customized properties are retained. The `oracle-apm` directory and the `oracle-apm-config.json` file in `$NODE_PATH` are backed up to `${STAGE_DIR}/temp`.

> **Note:**
>
> Review the `hostName` property, since this value will overwrite the `hostName` which was used by the agent prior to upgrade.
>
> The upgrade computes a default `hostname` and uses the same. To override this default, rerun the provisioning script with `ORACLE_HOSTNAME` argument.

The following changes occur when you choose to upgrade APM Node.js Agent:

1. A new version of APM Node.js Agent software will be installed in the `node-modules` folder.
2. Updated `oracle-apm-config.json` will be copied over from `STAGE_DIR` to `node-modules/oracle-apm/data` folder. Existing file will be backed up with a `.backup` extension. Custom properties edited previously like `proxy params` etc will be copied over to the new file.
3. All `.cer` certificate files will be converted into the correct format and copied over to `node-modules/oracle-apm/data` folder.
4. Start the Node.js applications. The required `config` files will be copied to the respective `NODE_APP_HOME/oracle-apm/data` folder.

Troubleshooting Upgrade Issues

The upgrade overwrites `oracle-apm` agent files in the `%NODE_PATH%` folder. If any file from this folder (for example, `oracle-apm-config.json` file) is open during the upgrade, the following `npm` error occurs, and the upgrade process stops.

```
npm ERR! code EPERM
npm ERR! errno -4048
npm ERR! syscall rename

npm ERR! Error: EPERM: operation not permitted, rename '%NODE_PATH%
\oracle-apm' -> '%NODE_PATH%\.oracle-apm.DELETE'
```

With the above error, the `oracle-apm` folder is deleted and hence, a rerun of the script is treated as a new install. Any locally modified properties are lost due to the rerun.

Workaround

Follow this workaround before attempting a rerun of the provisioning script:

1. The backup copy of the old `oracle-apm` folder is available in the backup folder - `${STAGE_DIR}/temp` (on Linux) and `%TEMP% folder` (on Windows). Copy this folder into `node_modules.`
2. Ensure that none of the files from the `node_modules/oracle-apm` folder are open, and rerun the provisioning script.

Remove APM Node.js Agent

To disable and remove APM Node.js Agents, complete the following tasks:

- Task 1: Remove the APM Node.js Agent from the application

- Task 2: Delete APM Node.js files from the Application folder
- Task 3: Uninstall APM Node.js Agent

Task 1: Remove the APM Node.js Agent from the application

1. Navigate to the `NODE_APP_HOME` folder.
2. Open the .js file that serves as the application's entry point.
3. In the .js file, search for the line that starts with `require('oracle-apm');` and delete it.

   Code with Node.js Agent enabled:

   ```
   require('oracle-apm'); //Newly added require for Oracle APM
   instrumentation

   var express = require('express');
   var app = express();

   app.get('/', function (req, res) {
     res.send('Hello World!');
   });

   app.listen(3000, function () {
     console.log('Example app listening on port 3000!');
   });
   ```

   Code after removing APM Node.js Agent

   ```
   var express = require('express');
   var app = express();

   app.get('/', function (req, res) {
     res.send('Hello World!');
   });

   app.listen(3000, function () {
     console.log('Example app listening on port 3000!');
   });
   ```

Task 2: Delete APM Node.js files from the Application folder

1. Navigate to the `NODE_APP_HOME` folder.
2. Remove the directory `oracle-apm` where APM Node.js Agent was installed.

   ```
   cd $NODE_APP_HOME
   rm -r oracle-apm
   ```

Task 3: Uninstall APM Node.js Agent

1. Set the environment variable PATH to include `$Node_Home/bin`.

2. Run the following command to uninstall APM Node.js Agent:

   ```
   npm uninstall -g oracle-apm
   ```

Uninstall APM Ruby Agent

Follow these steps to uninstall the APM Ruby Agent.

Remove APM Ruby Agent

1. Edit the application's Gemfile and either comment out or remove the gem reference:

   ```
   gem 'oracle_apm', '~> 1.x.x’
   ```

2. Restart your application server.
3. Remove the agent gem from the gem library:

   ```
   gem uninstall oracle_apm
   ```

4. To remove the Agent from the **Oracle Management Cloud UI**, in the **Oracle Management Cloud Agents** page, click **APM Agents** on the left navigation pane.
5. In the **APM Agents** page, select the APM Agent that you want to remove. Click the **Actions** menu and select **Remove**.

Disable and Remove Agents for Licensing Purposes

Charges for APM usage vary by licensing model.

In the Universal Credits licensing model, APM Agents are counted based on the amount of data received, and not on the number of registered APM Agents. In this licensing model, you simply need to stop data from being received by Oracle Management Cloud in order to stop or pause charging. You do not have to completely remove the APM Agents in this case. To stop data transfer, you can do ONE of the following:

- Disable APM data being sent to OMC: If you are using a Gateway, first disable End User Monitoring (EUM) (by setting the *End User Monitoring Injection Type Property* to None) using the APM Console and then stop the gateway agent. To deploy APM agents using gateways, refer to installation instructions for your type of agent. For detailed instructions on how to turn off EUM, see Enable and Configure End User Monitoring.
- Stop the Application Server. Data will no longer be sent to OMC.
- Remove APM Agents. For instructions on how to remove APM agents, see Upgrade APM Agents.

If you are using a Subscription-based licensing model the APM Agent count is based on the number of registered APM Agents. To stop charges related to APM Agent registrations:

- Stop, Disable and then Remove the APM Agents so that data is no longer sent to OMC. For instructions on how to remove APM agents, see Upgrade APM Agents.

A

Adding Application Classifications

You can add Classifications that can be used while creating Application Definitions.

Classifications help in grouping together related objects. Application Performance Monitoring uses the applied classifications to run filters. You can add Classifications while deploying APM.

To add a classification:

1. While deploying your APM Agent, add this property to `apmagent/config/AgentStartup.properties` file:

   ```
   oracle.apmaas.agent.appServer.classifications=<name of
   Classification>
   ```

   This classification can be used while creating application definitions.

   You can also add this property after deploying the APM Agent. Ensure that you restart the application server after editing the `properties` file.

The APM agent initially looks for the `AgentStartup.properties` file in the application server instance directories. If not found, the agent then uses the global `AgentStartup.properties` file.

B

Syntax for Using the APM Java Agent Installation Script

Here's the syntax for using the `ProvisionApmJavaAsAgent.sh` script:

```
ProvisionApmJavaAsAgent.sh -d  <domain_home> [ -h <hostname> ]
                        [ -no-prompt [-overwrite]] [ -no-wallet ]
                        [ -ph proxy_host> -pp proxy_port> [ -pt
proxy_auth_token> ] ]
                        [ -c client_collector_url> ] [ -regkey-file ]
[ -classifications ]
```

The following table describes the parameters of the `ProvisionApmJavaAsAgent.sh` script.

Parameters	Description
`-d <domain_home>`	Specify the absolute path of the home directory of your application server. This should be the absolute path of the home directory. The APM Java Agent software will be installed under this directory.
`-h <hostname>` (Optional)	Specify a valid fully qualified host name of your application server. By default, the installation script determines the hostname of the machine. Specify this parameter to override this value with another hostname.
`-no-prompt` (Optional)	*Do not prompt for confirmation.* Usually, the `ProvisionApmJavaAsAgent.sh` script displays various pieces of information that is either supplied to it, or that it derives. Further, the script asks you to confirm the values. When `-no-prompt` parameter is used, the values are displayed, but you will not be prompted before proceeding.
`-overwrite` (Optional)	*Deploy APM Agent and overwrite existing APM Agent, if any.* This parameter works with the `-no-prompt` parameter. • Specify `-overwrite` with `-no-prompt` to overwrite APM Java Agent installation without confirmation. • Use `-no-prompt` without `-overwrite`to upgrade the existing installation. With upgrade, all customized properties are retained.

Parameters	Description
`-no-wallet` (Optional)	*Do not use Oracle Wallet.* The `ProvisionApmJavaAsAgent.sh` script assumes that the `AgentInstall.sh` script specified an Oracle auto-login wallet containing the APM Java Agent's authorization token. This authorization token allows the APM Java Agent to contact the cloud services that it uses. Oracle Wallet is provisioned as the APM Java Agent's **credential store**. This option supports an environment which does not have the capability to use Oracle Wallet. If the `AgentInstall.sh` script does not specify an Oracle Wallet, then the provisioning script looks for the authorization token in the properties file that is also specified by the `AgentInstall.sh` script. If the `-no-wallet` parameter is specified, then while running the provisioning script, this flag ensures that the provisioning script does not use Oracle Wallet, even if one was provided by the `AgentInstall.sh` script. Instead, the authorization token of the APM Java Agent will be taken from the properties file specified by the `AgentInstall.sh` script, and provisioned in an alternative (non-wallet) credential store for the APM Java Agent to use.
`-ph <proxy host>` (Optional)	Specify the proxy host name if the APM Java Agent uses an HTTP proxy.
`-pp <proxy port>` (Optional)	Specify the proxy port if the APM Java Agent uses an HTTP proxy.
	Specify the proxy authorization token if the APM Java Agent is using HTTP proxy that requires authentication. The token is added to the credential store of the APM Java Agent. The token is either an Oracle *auto-login* wallet or the alternative credential store if a wallet is not being used.
	Specify the location of the file that contains your registration key. For more information on Registration Keys, go to Managing Registration keys in *Installing and Managing Oracle Management Cloud Agents*.
	Specify a classifications string that will be set in `AgentStartup.properties` file. This string is used to tag all requests related to a certain application.

Sample Command

Following is a sample command that is used to run the `ProvisionApmJavaAsAgent.sh` script.

```
./ProvisionApmJavaAsAgent.sh -d /WLS12.1.3/oracle_home/user_projects/
domains/acme_domain
```

Use of Outbound Proxy

Optionally, you can use an outbound proxy as the communication channel for the application. If your environment uses an outbound proxy, then when you provision your APM Java Agent, there are extra parameters in the `ProvisionApmJavaAsAgent.sh` script to specify the proxy host and port. For more information, run the `ProvisionApmJavaAsAgent.sh` script with the `-help` option.

Proxy Authentication

Use of proxy authentication is dependent on your environment. If you configured the proxy to require authentication, there are additional parameters you must specify while running the `ProvisionApmJavaAsAgent.sh` script. For more information, run the `ProvisionApmJavaAsAgent.sh` script with the `-help` option.