**Practical Exercise: Getting Started with WSO2 API Manager - Integration Profile**

Training Objective

Install and set up the Micro Integrator.

Business Objectives

Get an understanding of what constitutes the Micro Integrator.

High Level Steps

* Install Micro Integrator.
* Familiarize yourself with the product.

Detailed Instructions

**Install Micro Integrator**

Before installing the product, ensure that the installation prerequisites have been fulfilled.

# Installation Prerequisites

WSO2 API Manager contains multiple runtimes. Before installing the runtimes, be sure that the appropriate prerequisites are fulfilled.

## Environment compatibility

Listed below are the OS and database requirements.

|  |  |
| --- | --- |
| Operating systems\ Databases | * Install a JDK version that is OpenJDK 8,11 or Oracle JDK 8,11. * All WSO2 Carbon-based products are generally compatible with most common DBMSs. The embedded H2 database is suitable for development, testing, and some production environments. For most enterprise production environments. However, WSO2 recommends that you use an industry-standard RDBMS such as Oracle, PostgreSQL, MySQL, MS SQL, etc. Additionally, WSO2 does not recommend the H2 database as a user store. * It is **not recommended to use Apache DS** in a production environment due to scalability issues. Instead, use an LDAP like OpenLDAP for user management. * On a production deployment, it is recommended that WSO2 products are installed on the latest releases of RedHat Enterprise Linux or Ubuntu Server LTS. |

## System requirements

Check the following system requirements for the Micro Integrator runtimes.

### Micro Integrator runtime

|  |  |
| --- | --- |
| **Type** | **Requirement** |
| **Docker** | * ~512 MB heap size for one Micro Integrator instance. This is generally sufficient for processing typical SOAP messages. However, the requirements vary with larger message sizes and the number of messages processed concurrently. * 1 GB memory for a Docker container. * Minimum 0.5 core per Micro Integrator Docker instance. |
| **Virtual Machine (VM)/Physical** | * Minimum 0.5 core (1.0-1.2 GHz Opteron/Xeon processor). * 1 GB RAM for JVM. * ~512 MB heap size. This is generally sufficient for processing typical SOAP messages. However, the requirements vary with larger message sizes and the number of messages processed concurrently. |

## Database storage requirements

Check the following database storage requirements for the API-M runtime.

### API-M runtime

|  |  |  |
| --- | --- | --- |
| **DB Type** | **Minimum storage required for databases** | |
| **AM Database** | **Shared Database** |
| **MySQL** | 1 GB | 1 GB |
| **Oracle** | 1 GB | 1 GB |
| **MSSQL** | 1 GB | 1 GB |
| **Oracle RAC** | 1 GB | 1 GB |
| **PostgreSQL** | 1 GB | 1 GB |
| **IBM DB2** | 1 GB | 1 GB |
| **MariaDB** | 1 GB | 1 GB |

## Required applications

The following applications are required for running WSO2 API Manager and its samples.

|  |  |  |  |
| --- | --- | --- | --- |
| **Application** | **Purpose** | **Version** | **Download Links** |
| **Apache Ant** | * To compile and run the product samples in <API-M\_HOME>/samples. | 1.7.0 or later | [Apache Ant](http://ant.apache.org/) |
| **Web Browser** | * The Web browser must be JavaScript enabled to take full advantage of user interfaces.   **Note**  On Windows Server 2003, you must not go below the medium security level in Internet Explorer 6.x. |  |  |

# Installing the Micro Integrator Runtime

Follow the steps given below to install the Micro Integrator (MI) runtime of WSO2 API Manager.

## Before you begin

Java Development Kit (JDK) is essential to run the product.

## Installing the Micro Integrator

1. Go to the [WSO2 Micro Integrator web page](https://wso2.com/integration/micro-integrator/), click **Download**, and then click **Zip Archive** to download the Micro Integrator distribution as a ZIP file.
2. Extract the archive file to a dedicated directory for the Micro Integrator, which will hereafter be referred to as <MI\_HOME>.

## Setting up JAVA\_HOME

You must set your JAVA\_HOME environment variable to point to the directory where the Java Development Kit (JDK) is installed on the computer.

**Info**

Environment variables are global system variables accessible by all the processes running under the operating system.

On Linux/OS X

1. In your home directory, open the BASHRC file (.bash\_profile file on Mac) using editors such as vi, emacs, pico, or mcedit.
2. Assuming you have JDK 11 in your system, add the following two lines at the bottom of the file, replacing /usr/java/jdk-11.0.x with the actual directory where the JDK is installed.

On Linux:

export JAVA\_HOME=/usr/java/jdk-11.0.x

export PATH=${JAVA\_HOME}/bin:${PATH}

On OS X:

export JAVA\_HOME=/System/Library/Java/JavaVirtualMachines/11.0.x.jdk/Contents/Home

1. Save the file.

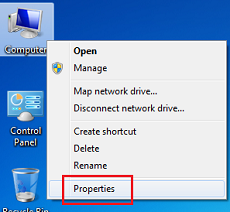
On Windows

Typically, the JDK is installed in a directory under C:/Program Files/Java , such as C:/Program Files/Java/jdk-11.0.x. If you have multiple versions installed, choose the latest one, which you can find by sorting by date.

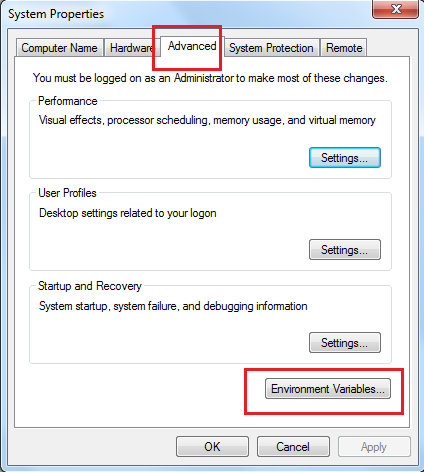
You set up JAVA\_HOME using the System Properties, as described below. Alternatively, if you just want to set JAVA\_HOME temporarily for the current command prompt window, set it at the command prompt.

**Setting up JAVA\_HOME using the system properties**

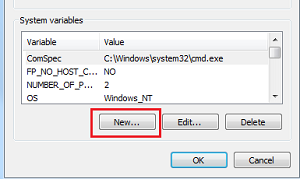
1. Right-click the **My Computer** icon on the desktop and click **Properties.**



1. In the System Properties window, click the **Advanced** tab, and then click **Environment Variables**.



1. Click **New** under **System variables** (for all users) or under **User variables** (just for the user who is currently logged in).



1. Enter the following information:

* In the **Variable name** field, enter: JAVA\_HOME
* In the **Variable value** field, enter the installation path of the Java Development Kit, such as: c:/Program Files/Java/jdk-11.0.x

The JAVA\_HOME variable is now set and will apply to any subsequent command prompt windows you open. If you have existing command prompt windows running, you must close and reopen them for the JAVA\_HOME variable to take effect, or manually set the JAVA\_HOME variable in those command prompt windows as described in the next section. To verify that the JAVA\_HOME variable is set correctly, open a command window (from the **Start** menu, click **Run**, and then type CMD and click **Enter**) and execute the following command:

set JAVA\_HOME

The system returns the JDK installation path.

## Setting system properties

If you need to set additional system properties when the server starts, you can take the following approaches:

* **Set the properties from a script** : Setting your system properties in the startup script is ideal because it ensures that you set the properties every time you start the server. To avoid having to modify the script each time you upgrade, the best approach is to create your startup script that wraps the WSO2 startup script and adds the properties you want to set, rather than editing the WSO2 startup script directly.
* **Set the properties from an external registry** : If you want to access properties from an external registry, you could create Java code that reads the properties at runtime from that registry. Be sure to store sensitive data such as username and password to connect to the registry in a property file instead of in the Java code and secure the properties file with the [secure vault](https://docs.wso2.com/display/ADMIN44x/Carbon+Secure+Vault+Implementation).