

AppDynamics Universal Agent

On this page:

- [About the Universal Agent](#)
- [Set Up Tasks](#)
- [Getting Started](#)

Search Universal Agent topics:



✱

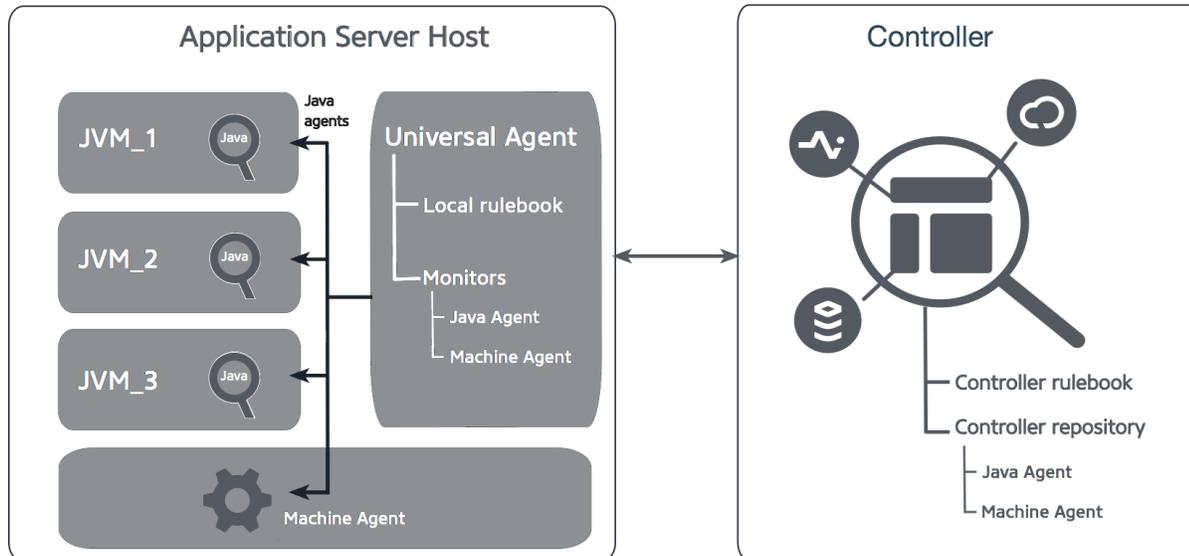
The Universal Agent provides a central management point for administering runtime agents for diverse application environments and eases the management of large environments. The Universal Agent automates the tasks of installing, running, and maintaining AppDynamics runtime agents, such as the Machine Agent and Java Agent.

After you setup the Universal Agent, you can update the versions of the runtime agents by putting new distributions in a central repository and updating the configuration with the latest version. To further ease agent management in large and dynamic environments, the Universal Agent configuration supports remote JVM attachment and dynamic node naming. This topic describes how to set up and use the Universal Agent.

About the Universal Agent

The Universal Agent enables you to run and manage AppDynamics runtime agents by using a single configuration file, called a *rulebook*. The rulebook determines whether the Universal Agent installs, starts, and stops runtime agents. By default, the Universal Agent runs in *controller mode* and applies a shared rulebook stored and served by the Controller. Alternatively, you can run the Universal Agent in *local mode*, in which it uses a rulebook (stored in the Universal Agent home directory) that is specific to the instance of the agent. See [Universal Agent Rulebooks](#) for more details on operation modes.

When the Universal Agent finds a new rule for a runtime agent in the rulebook, it retrieves the runtime agent from a shared repository, installs the agent as a local monitor, and starts the agent. You can manage multiple runtime agents on a single host, which allows for having different versions of the runtime agents for each JVM or each host in your environment.



Set Up Tasks

To use the Universal Agent, you need to do the following tasks:

1. [Set up](#) the runtime agent source repository as either a Controller repository or file system repository. A file system repository would typically be on a shared repository accessible to the Universal Agent host machines.
2. [Install](#) the Universal Agent on the machines to be monitored.
3. [Create](#) the rulebooks to manage your runtime agent instances.

Getting Started

The following procedures walk you through the three set-up tasks for a simple scenario using a single Linux-based application server with an on-premises Controller. On Linux, these steps require `sudo` access to the application server machine, because the setup establishes the Universal Agent as a service.

When following these steps, adjust the values in the commands for your specific Controller, user, account, and so on. Before starting, review additional information in [Install the Universal Agent](#).

Set up the runtime agent source repository

This procedure sets up a Controller source repository.

1. Confirm that you have an on-premises Controller installed and running and you have access to the Controller machine.
2. On the controller, create the `<controller-home>/agent-binaries` directory. This directory serves as your Controller source repository for the runtime agent binaries.
3. Put the Universal Agent ZIP file in the directory.
4. Rename it to `universalagent-<version>-64bit-linux.zip`.
5. Download the Machine Agent ZIP file to the Controller source repository, `<controller-home>/agent_binaries` directory.
6. Rename the Machine Agent file to `machine-<version>-64bit-linux.zip`.
7. Download the Java Agent ZIP to the Controller source repository, `<controller-home>/agent_binaries` directory.
8. Rename the Java Agent file to `java-<version>.zip`. (Note there is no bitness in this name.)
9. Continue to the next section to install the Universal Agent.

Install the Universal Agent on the machines to be monitored

1. Get the account key for your Controller:
 - a. From the Controller UI, open the License page and click the **Accounts** tab.
 - b. Click the link to show the key.
2. From the application server machine, get the Universal Agent distribution. You can use the following CURL command to get the Universal Agent from the Controller repository.
Replace the <controller_user>, <controller_account>, <controller_password>, and <controller_host> address and <primary_port> with values for your system. The Controller user should be an account in the Controller UI that has Universal Agent administration permissions. Replace <account_access_key> with your key and use the version of your specific Universal Agent.

```
curl -u '<controller-user>@<controller-account>:<controller-password>' -X POST -d
'action=downloadAgent&agentVersion=<version>&agentName=universalagent&classifier=64bit-linux' -
o ua<version>.zip 'http://<controller-host:primary-port>/controller/FileDownloadServlet'
mkdir ua_install
unzip ua<version>.zip -d ua_install/
sudo ./ua-install/ua<version>/bin/install.sh --account-access_key <account-access_key>
```

By default, the script installs the agent in the directory, /opt/appdynamics/universal-agent/. You can change that using the argument --target, but the steps (below) for creating your rulebooks assume the default installation location, /opt.

Create the rulebook

1. From the application server machine, where you installed the Universal Agent, execute a REST API command against your Controller to add a single default rulebook.
Replace <controller-user>, <controller-account>, <controller-password>, <controller-host>, and <controller-port> with the values for your Controller.

```

curl -i -X PUT -su '<controller-user>@<controller-account>:<controller-password>' -H "Content-
type: application/json" -H "Accept:application/json" \
    "http://<controller-host>:<controller-port>/controller/universalagent/v1/user/rulebooks
/byName/default-controller" --data '{
    "name": "default-controller",
    "comments": "An example rule book to monitor Java Agent, Machine Agent and Universal
Agent itself.",
    "rules": [
        {
            "config": {
                "state": "started",
                "version": "<version>"
            },
            "monitor": "machine",
            "comments": "Rule to monitor Machine Agent",
            "condition": "True",
            "name": "Machine monitor"
        },
        {
            "name": "Universal Agent rule",
            "comments": "Rule to monitor Universal Agent",
            "monitor": "universal",
            "config": {
                "version": "<version>",
                "state": "started"
            },
            "condition": "True"
        },
        {
            "config": {
                "version": "<version>",
                "state": "started",
                "application_name": "YourAppName",
                "tier_name": "1stTier"
            },
            "condition": "True"
            "name": "Java Monitor",
            "comments": "Rule to monitor Java Agent",
            "monitor": "java"
        }
    ]
}'

```

2. After a few moments, verify that the Universal Agent retrieves and installs the Machine Agent and the Java Agent. The corresponding agent files should now appear in the following directories:
 - Machine Agent: /opt/appdynamics/universal-agent/monitor/machine
 - Java Agent: /opt/appdynamics/universal-agent/monitor/java
 - or <target_dir>/appdynamics/universal-agent/monitor/... if you specified --target <target_dir> in the installation step
3. Check the status of the Universal Agent as reported to the Controller using the REST API, universalagent/v1/user/agents/summary. For instance, using cURL, enter the following command: Replace the username, password, and Controller address with values appropriate for your environment. The response should list the running agents with the version number and the applied rulebook for each.

```

curl -s -X GET -u '<controller-user>@<controller-account>:<controller-password>' -H 'Content-
type: application/json' http://<controller-host>:<controller-port>/controller/universalagent/v1
/user/agents/summary

```