

# **Arm Licence Server User Guide**

Version 19.1

The Arm logo, consisting of the lowercase letters 'arm' in a bold, blue, sans-serif font.

# Contents

<b>Contents</b>	<b>1</b>
<b>1 Introduction</b>	<b>2</b>
1.1 Online resources . . . . .	2
1.2 Obtaining help . . . . .	2
<b>2 Installation</b>	<b>3</b>
2.1 Linux installation . . . . .	3
<b>3 Using Arm Licence Server</b>	<b>4</b>
3.1 Running Arm Licence Server . . . . .	4
3.1.1 Systemd-based distributions . . . . .	4
3.1.2 SysVinit-based distributions . . . . .	5
3.2 Logging . . . . .	5
3.3 Adding a new license . . . . .	6
3.4 Example . . . . .	6
3.5 Running product clients . . . . .	7
3.6 Architecture licensing . . . . .	8
3.6.1 Using multiple architecture licenses . . . . .	8
3.7 Accessing Arm Licence Server behind a firewall . . . . .	9
3.8 Querying status . . . . .	9
3.9 Handling lost clients . . . . .	10
3.10 Troubleshooting . . . . .	10

# Introduction

Welcome to the Arm Licence Server user guide. This guide covers installation and use of Arm Licence Server.

There are several license types available for Arm products. These can be summarized as:

- Evaluation licenses
- Workstation licenses
- Supercomputing cluster licenses

For the case where evaluation and workstation licenses are used, the license file is simply copied to the Arm product installation directory, `{installation-dir}/licences`, and Arm Licence Server is not required.

In the case of supercomputing cluster scenarios where evaluation licenses are not being used, the use of Arm Licence Server is a requirement.

Note that when the remote clients for Mac OS X , Windows and Linux are used, the license file in this case is installed on the target system that the remote client connects to for the purposes of debugging or analysis. This applies whether or not an evaluation license is being used. For example, a remote client for Mac OS X needs to debug a remote system for evaluation purposes. In this case the evaluation license needs to be installed on the system to be debugged, and not where the remote client is installed. Once the evaluation period ends, new licenses should be purchased, and Arm Licence Server should be installed to handle licensing.

Installation of licenses is discussed further in the [Adding a new license](#) section.

Once you have determined that Arm Licence Server is a requirement for your site, you can proceed with [Installation](#).

## Online resources

You can find links to tutorials, training material, webinars and white papers in our online knowledge center:

**Knowledge Center** [Arm help and tutorials](#)

Known issues and the latest version of this user guide may be found on the support web pages:

**Support** [Arm Developer website](#)

## Obtaining help

You can contact Arm support at [Arm support](#).

# Installation

A release of Arm Licence Server may be downloaded from the [Arm Developer website](#).

## Linux installation

Arm Licence Server is only available for Linux systems. The install script `textinstall.sh` should be used:

```
tar xf arm-licence server-19.1-<distro>-<arch>.tar  
cd arm-licence server-19.1-<distro>-<arch>  
./textinstall.sh
```

replacing `<distro>` and `<arch>` with the OS distribution and architecture of your tar package, respectively. For example, the tarball package for Redhat 7.4 OS and Armv8-A (AArch64) architecture is: `arm-licence server-19.1-Redhat-7.4-aarch64.tar`

Press *Return* to read the license when prompted and then enter the directory where you would like to install Arm Licence Server.

Alternatively, to run the text-mode install script `textinstall.sh`, accept the license, and point to an installation directory in one step, pass the arguments `--accept-licence` and `<installation_directory>` when executing `textinstall.sh`. For example:

```
./textinstall.sh --accept-licence <installation_directory>
```

replacing the `<installation_directory>` with a directory of your choice.

## Using Arm Licence Server

Arm Licence Server is capable of serving several different types of license, enabling one server to serve all Arm software in an organization.

It should be pointed out that Arm Licence Server is not required to serve evaluation licenses and workstation licenses, only floating licenses for HPC clusters.

## Running Arm Licence Server

To ensure security, Arm Licence Server should only be run as an unprivileged user. For example, `no-body`, or preferably a dedicated user.

If run without arguments, Arm Licence Server uses licenses in the current directory, that is, files matching `Licence*` and `License*`. An optional argument specifies the directory to be used instead of the current directory.

Typically floating license server files are stored in the Arm Licence Server installation directory, for example, `/opt/arm/licenceserver/licenses`. This is also the directory the `systemd` unit and Sys-V style `init` script use by default.

System administrators typically configure the license server to start automatically during system boot. Arm Licence Server includes a `systemd` unit configuration file for Red Hat Enterprise Linux 7 and SUSE Linux Enterprise Server 12, and a SysV-style `init` script for Debian, Ubuntu, and previous Red Hat and SUSE releases.

Before configuring Arm Licence Server to start automatically, an unprivileged user should be created, and a log directory allocated, as shown in the following example (these commands must be run as `root`):

```
adduser --system --user-group --no-create-home \
        --home-dir /opt/arm/licenceserver allinea
mkdir /var/log/allinea
chown allinea: /var/log/allinea
```

This creates the user `allinea`, creates the log directory `/var/log/allinea`, and finally the user is assigned ownership of the log directory.

Note that `/opt/arm/licenceserver` is the location of the Arm Licence Server installation.

The license server log file is `/var/log/allinea/allinea.log` by default.

## Systemd-based distributions

This section shows you how to run Arm Licence Server on `Systemd`-based systems.

For Red Hat Enterprise Linux 7 and SUSE Linux Enterprise Server 12, a `systemd` unit configuration file can be used to start Arm Licence Server. To use it, follow the procedure below (as `root`):

1. Edit `lib/systemd/system/allinea-licenceserver.service` in your Arm Licence Server installation. Make sure that the `Environment`, `ExecStart` and `User` settings in the `[Service]` section are correct for your system.
2. Enable the license server on future boots:

```
systemctl enable \
lib/systemd/system/allinea-licenceserver.service
```

3. Start the license server:

```
systemctl start allinea-licenceserver
```

4. Check that it started properly:

```
systemctl status allinea-licenceserver
```

## SysVinit-based distributions

This section shows you how to run Arm Licence Server on SysVinit-based systems.

`bin/allinea\_licensing\_init` is a SysV-style `init` script for both Debian and Ubuntu, and previous Red Hat Enterprise Linux and SUSE Linux Enterprise Server releases. To use the script, follow the procedure below (as `root`):

1. Set `ALLINEA_TOOLS_PATH` at the top of the script to point to the Arm Licence Server installation, for example, `/opt/arm/licenceserver`.
2. Set `ALLINEA_LICENCE_USER` to the name of the user to run the license server as. This should be the user already created for this purpose, for example, `allinea`.
3. Install the script to `/etc/init.d`:

```
ln -s /opt/arm/licenceserver/bin/allinea_licensing_init \
/etc/init.d/
```

4. Enable the service at boot time:

```
# On Red Hat or SUSE systems:
chkconfig --add allinea_licensing_init

# On Debian or Ubuntu systems:
update-rc.d allinea_licensing_init defaults
```

5. Start the service:

```
service allinea_licensing_init start
```

6. Check that it started correctly:

```
service allinea_licensing_init status
```

## Logging

This section shows you how to enable logging.

Set the environment variable `ALLINEA_LICENCE_LOGFILE` to the file that you wish to append log information to.

Set `ALLINEA_LICENCE_LOGLEVEL` to specify the amount of information required. These steps must be done prior to starting the server.

The following list specifies the available log levels:

- Level 0: no logging.
- Level 1: client licenses issued are shown, served licenses are listed.
- Level 2: stale licenses are shown when removed, licenses still being served are listed if there is no spare license.
- Level 3: full request strings received are displayed.
- Level 6: is the maximum.

In level 1 and above, the MAC address, user name, process ID, and IP address of the clients are logged.

## Adding a new license

This section shows you how to add a floating license to Arm Licence Server.

A floating license consists of two files: the server license (a file named `Licence.xxxx`) and a client license file `Licence`.

The following procedure demonstrates how to add a new floating license:

- Copy the server license file to the `licences` subdirectory of the Arm Licence Server installation. For example, `/opt/arm/licenceserver/licences`.
- Copy the client license file to the `licences` subdirectory of the installation of the licensed product, `{forge-or-reports-installation-directory}/licences`. For example, `/opt/arm/forge/licences/Licence`. The client license file needs to be edited to specify the hostname of the license server being used.
- Restart Arm Licence Server for the new floating license to be picked up. Existing clients should not experience disruption if the restart is completed promptly.

It is permissible to have multiple floating licenses being served from the same `licences` directory. For example, you may have one server license file for Arm DDT and another for Arm Performance Reports. If you have multiple licenses for the same product the license with the most tokens will be preferred.

*Note: Workstation and Evaluation license files do not need a license server and should be copied directly to the Arm Forge or Arm Performance Reports `licences` subdirectory, for example, `/home/user/arm/forge/licences/Licence.ddt`.*

## Example

In this example it is assumed that there is a dedicated license server machine running Red Hat Enterprise Linux 6. The Arm Licence Server is installed in `/opt/arm/licenceserver` and the server license files are stored in `/opt/arm/licenceserver/licences`.

*Note: if you install the Arm Licence Server to a different installation path you will need to edit the `ALLINEA_TOOLS_PATH` variable in the `allinea_licensing_init` script.*

To run Arm Licence Server program as the dedicated `allinea` user, serving all licenses in `/opt/arm/licenceserver/licences`, and logging most events to the `/var/log/allinea/allinea.log`, use the following procedure:

```
$ su - root
% adduser --system --user-group --no-create-home \
    --home-dir /opt/arm/licenceserver allinea
% mkdir /var/log/allinea
```

```
% chown allinea: /var/log/allinea
% ln -s /opt/arm/licenceserver/bin/allinea_licensing_init \
  /etc/init.d/
% chkconfig --add allinea_licensing_init
% service allinea_licensing_init start
```

If the server license file is `/opt/arm/licenceserver/licences/Licence.server.physics` and is served by the machine `server.physics.acme.edu`, on port 4252, the server license file would resemble the following:

```
type=3
serial_number=1014
max_processes=48
expires=2004-04-01 00:00:00
support_expires=2004-04-01 00:00:00
mac=00:E0:81:03:6C:DB
interface=eth0
debuggers=gdb
serverport=4252
max_users=2
beat=60
retry_limit=4
hash2=c18101680ae9f8863266d4aa7544de58562ea858
```

Then the client license could be stored at `/opt/arm/forgelicences/Licence.client.physics` and contain:

```
type=2
serial_number=1014
hostname=server.physics.acme.edu
serverport=4252
```

Note that the client file `hostname` parameter has been edited to reference Arm Licence Server.

## Running product clients

For each floating license the corresponding client license file should be copied to `licences` subdirectory of the installation of the particular product, `{forge-or-reports-installation-directory}/licences`, for example, `/opt/arm/forgelicences/Licence`). You will need to edit the `hostname` line to contain the host name or IP address of the machine running the Arm Licence Server.

You may specify an alternative location of the license directory for the product clients by setting an environment variable, `ALLINEA_LICENCE_DIR`, before they are launched.

The environment variable may be set, in a *module* file for the product. For example:

```
export ALLINEA_LICENCE_DIR=${HOME}/SomeOtherLicencedir
forge
```

The client license file is in plain text and may be modified as required.

The fields in the client license file are:

Name	Required	Description
------	----------	-------------



hostname	Yes	The hostname, or IP address of the license server
ports	No	A comma separated list of ports to be tried locally for GUI-backend communication, Defaults to 4242,4243,4244,4244,4245
serial_number	Yes	The serial number of the server license to be used
serverport	Yes	The port the server listens on
type	Yes	Must have value 2—this identifies the license as needing a server to run properly

*Note: The serial number of the server license is specified as this enables a user to be tied to a particular license.*

## Architecture licensing

Licenses issued after the release of Arm Licence Server 6.1 specify the compute node architectures that they may be used with. Licences issued prior to this release will enable the x86\_64 architecture by default. Existing users for other architectures will be supplied with new licenses that will enable their architectures.

If there is any problem then contact Arm support at [Arm support](#).

## Using multiple architecture licenses

If you are using multiple license files to specify multiple architectures, Arm recommends that you follow these steps.

### Procedure

1. Ensure that the default licenses directory is empty.
2. Create a directory for each architecture.
3. When you want to target a specific architecture, set `ALLINEA_LICENSE_DIR` to the relevant directory.  
Alternatively, set `ALLINEA_LICENSE_FILE` to specify the license file.

### Example

On a site that targets two architectures, x86\_64 and AArch64, create a directory for each architecture, and name them `licenses_x86_64` and `licenses_aarch64`. Then, to target the architectures, set the license directories as follows:

To target AArch64:

```
export ALLINEA_LICENSE_DIR=/path/to/licenses/licenses_aarch64
```

To target x86\_64:

```
export ALLINEA_LICENSE_DIR=/path/to/licenses/licenses_x86_64
```

## Accessing Arm Licence Server behind a firewall

In some scenarios Arm Licence Server may be located behind a firewall. This is the case if you are accessing a license server hosted by Arm. You may also be debugging a system that is not on the same network as the license server. In this case SSH forwarding can be used to access the license server that is behind the firewall.

A local client license file is created or modified to specify `localhost` as the `hostname` parameter:

```
type=2
serial_number=1014
hostname=localhost
serverport=4252
```

Communication is then forwarded over the secure SSH connection to Arm Licence Server, which is listening on port 4252.

An example of the command to set up the SSH forwarding is shown here:

```
ssh -C -L 4252:server.physics.acme.edu:4242 login.physics.acme.edu
```

The `-C` switch is optional, and enables compression for the situation where communication takes place over relatively slow links.

## Querying status

Arm Licence Server provides a simple HTML interface to allow for querying of the current state of the licenses being served. This can be accessed in a web browser at the following URL:

```
http://<hostname>:<serverport>/status.html
```

For example, using the values described in the section [Example](#), the URL would be:

```
http://server.physics.acme.edu:4252/status.html
```

Initially, no licenses are being served, and the output in your browser window would be similar to the following:

```
[Licences start]
[Licence Serial Number: 1014]
[No licences allocated - 2 available]
[Licences end]
```

You can see that two licenses are available in this example.

As licenses are served and released, the information displayed will change. To update the status display, simply refresh your web browser window. For example, after one Arm product is started, the output is updated:

```
[Licences start]
[Licence Serial Number: 1014]
[1 licences available]
[Client 1]
[mac=00:04:23:99:79:65; uname=gwh; pid=14007; licence=1014]
[Latest heartbeat: 2004-04-13 11:59:15]
[Licences end]
```

Then, after another Arm product is started and the web browser window is refreshed (notice the value for number of licences available), the output is updated:

```
[Licences start]
[Licence Serial Number: 1014]
[0 licences available]
[Client 1]
[mac=00:04:23:99:79:65; uname=gwh; pid=14007; licence=1014]
[Latest heartbeat: 2004-04-13 12:04:15]
[Client 2]
[mac=00:40:F4:6C:4A:71; uname=graham; pid=3700; licence
=1014]
[Latest heartbeat: 2004-04-13 12:04:59]
[Licences end]
```

Finally, after the first Arm product finishes:

```
[Licences start]
[Licence Serial Number: 1014]
[1 licences available]
[Client 1]
[mac=00:40:F4:6C:4A:71; uname=graham; pid=3700; licence
=1014]
[Latest heartbeat: 2004-04-13 12:07:59]
[Licences end]
```

## Handling lost clients

Should Arm Licence Server lose communication with an instance of a client, the license allocated to that client is made unavailable for new clients until a timeout period has expired. The length of this timeout period can be calculated from the license server file values for `beat` and `retry_limit`:

$$\text{lost\_client\_timeout\_period} = (\text{beat seconds}) * (\text{retry\_limit} + 1)$$

For the example license files above, the timeout period would be:

$$60 * (4 + 1) = 300 \text{ seconds}$$

During this timeout period, details of the ‘lost’ client continue to be output by the status display. As long as additional licenses are available, new clients can be started. However, once all of these additional licenses have been allocated, new clients are refused a license while this timeout period is active.

After this timeout period has expired, the status continues to display details of the ‘lost’ client until another client is started. Arm Licence Server grants a license to the new client and the status display then reflects the details of the new client.

## Troubleshooting

Licenses are plain-text which enables the user to see the parameters that are set. A checksum verifies the validity of the license.

If problems arise, the first step is to check the parameters specified in the license file are consistent with the machine that is being used (MAC and IP address), and that the number of users is as expected.

# Index

Architecture licensing, [8](#)  
    Multiple licenses, [8](#)

Distributions  
    Systemd, [4](#)  
    SysVinit, [5](#)

Example, [6](#)

Firewall, [9](#)

Installation, [3](#)  
    Linux, [3](#)

Licence Server  
    Running, [4](#)  
    Using, [4](#)

License  
    Adding, [6](#)

Logging, [5](#)

Lost clients, [10](#)

Obtaining help, [2](#)

Online resources, [2](#)

Product clients, [7](#)

Status, [9](#)

Troubleshooting, [10](#)