Migrating your server to a new version

Newer server versions can often require a different server database or configuration. For example, if you have version 1.6 installed and you want to run version 1.9, your server requires a different configuration.

How migration works

The `setup` command detects your current server version and copies your server data to the `LicensingServerBackups` directory before migration begins. The location of this directory depends on the server's operating system and the version of the licensing server you're migrating to:

- **Linux**:
  - Version 2.0 or later: `/usr/share/unity3d/LicensingServerBackups`
  - Version 1.x: `~/.config/unity3d/Unity/LicensingServerBackups`
- **Windows**:
  - Version 2.0 or later: `%PROGRAMDATA%\Unity\LicensingServerBackups`
  - Version 1.x: `%LOCALAPPDATA%\Unity\LicensingServerBackups`

The migration continues by converting your data to be compatible with the new server version. When the migration is complete, it begins the configuration process, using the information you provided in the previous setup of the server.

> ! If the command encounters any problems during the migration, it cancels the migration and restores your data to its state before the migration.

Before you begin

Before starting any licensing server migration, be sure to:

- Plan for downtime. Depending on the current and target versions, migrating can take several minutes. For example, in Unity's labs, migrating from a version before 1.11.0 to a version later than 1.11.0 took 15 minutes to complete on a 3 GB database.
- As part of your downtime, factor additional time, in case you need to abandon the migration and revert to the original version.
- Identify the folder where your current server is installed, and identify a separate location for the new server. Don't overwrite the older version with the newer version, otherwise you can't easily revert the upgrade.
- Decide how to control administrator access. If you're migrating from version 1.x, you might have an allow list that used IP addresses to control administrator access. If you're migrating to version 2.0 or later, you have two options for controlling administrator access:
  - To keep the current system that uses IP addresses, you don't need to take any explicit steps. Any IP addresses in the allow list can continue to access the `admin` endpoint of API and the administrator mode of the dashboard. When users with these IP addresses access the dashboard, they will automatically log in to the administrator mode, bypassing the login screen.
  - To use credentials instead of IP addresses to control administrator access, you need to take extra steps. These steps are included in the procedure that follows.

Procedure

1. Download the setup file for the new server version. Refer to Download the server setup.

   > ⚠ Unzip the downloaded file's contents to a location other than the existing server. Placing the downloaded file's contents in a separate location ensures that you can revert to the existing server, if needed.

2. If your server runs as a service, stop the Licensing Server service.
3. From a Linux Terminal or Windows PowerShell, go to the Unity Licensing Server launch directory and run the following command:

   - **Linux**: `./Unity.Licensing.Server`
   - **Windows**: `.\Unity.Licensing.Server.exe`

   The server checks the `serverVersion` property in the server configuration file.

   If the server starts normally, skip to step 6.2.

   If incompatibility is detected, the following message displays, and the server stops.

   ```
   Cannot start the licensing server.
   Incompatible version is already installed.
   Please, make sure to stop existing Licensing.Server
   service (if it is running) and run setup command.
   ```

4. From a Linux Terminal or Windows PowerShell (as administrator), run the following command:
   - **Linux**: `sudo ./Unity.Licensing.Server setup`
   - **Windows**: `.\Unity.Licensing.Server.exe setup`

   > ⚠ Migrations that are complex will prompt for confirmation before proceeding.

   > ❗ The migration process writes detailed information to the server's log file, but it can take several minutes for new information to appear.

5. If you don't encounter any issues running the setup command, proceed to the next step. If you encounter issues that require you to abandon the migration and revert to the original version, skip the remaining steps and refer to Recovering from a failed migration.
6. When the setup completes, perform a final check:
   1. Start the server manually.
   2. Access one of the server endpoints to make sure the `serverStatus` is **Healthy**. For more information, refer to Testing the licensing server.
   3. Stop the server.
7. If your server runs as a service, you need to recreate the service. For details, refer to Set up a service.
8. If you're migrating from v1.x to 2.x, and you want to control administrator access using credentials exclusively (instead of IP addresses), follow these steps:
   1. Locate the licensing server configuration file for the new version of your server. For version 2.x, the full path to this file varies by platform:

        - **Linux**: `/usr/share/unity3d/LicensingServer/config/licensing-server-config.json`
        - **Windows**: `%PROGRAMDATA%\Unity\LicensingServer\config\licensing-server-config.json`
    2. Edit the file by removing the `adminIpAllowlist` key-pair value.
    3. Save the file.
9. Start the service, if applicable. Refer to Restarting the service.

Sample output

The following sample shows output from a migration that requires confirmation (Windows):

```
> .\Unity.Licensing.Server.exe setup
----
Welcome to Unity Licensing Server setup command line interface.
This setup will help you configure your license server
and generate server registration request file.
----

Upgrading from 1.10.0 to 1.11.0...

This migration can take more than a few minutes. Are you sure you want to run it now?

Proceeding with migration...
Backup installed version 1.10.0...Done
Scripts to run: 1.11.0
Executing 1.11.0...Done
Stamping version 1.11.0...Done
Successfully migrated from 1.10.0 to 1.11.0
Enter the server name (e.g. LicenseServer):
```

Recovering from a failed migration

When you migrate your server to a higher version, the `setup` command automatically creates a backup copy of your database file.

If the migration fails, the `setup` attempts to restore the backup. But if the `setup` fails to restore the backup, the following critical error message appears:

```
An error occurred while restoring a backup.
```

This critical error message indicates that your server is in an invalid state which might prevent you from continuing to use your previous server version.

To fix this error, or to roll back your migration for any other reason, copy the contents of the `LicensingServerBackups/<your_old_server_version>/` directory to the `LicensingServer` directory to manually restore the backup data. For example:

```
├── LicensingServer
  │   ├── config
  │   │   ├── key-b4573620-ca22-423f-8d52-e3a767391234.xml
  │   │   ├── licensing-server-config.json
  │   │   └── unity.licensing.delegation.key
  │   ├── data
  │   └── wwwroot\v1\package\acl
  │
  ├── LicensingServerBackups
  │   └── 1.6.0
  │       ├── config
  │       ├── data
  │       └── wwwroot
```

To restore the server to a previous version, follow these steps:

1. Put the existing database into the target folder, which varies depending on the server's operating system and the version of your licensing server:
   - **Linux**:
     - Version 2.0 or later: `/usr/share/unity3d/LicensingServer/data/`
     - Version 1.x: `~/.config/unity3d/Unity/LicensingServer/data/`
   - **Windows**:
     - Version 2.0 or later: `%PROGRAMDATA%\Unity\LicensingServer\data\`
     - Version 1.x: `%LOCALAPPDATA%\Unity\LicensingServer\data\`
2. Rename the file to `LicensingServerDb.sqlite`.
3. If your server runs as a service, and you created a service for the newer server, make sure the path to the server binary file is correct.

   If the service is defined to run the wrong version of the server, recreate the service using the binaries for the server version you want to use. For details, refer to Set up a service.
4. Restart the service, if applicable.

Additional resources#

- Running setup command (version 2.0 or later)
- Running setup command (version 1.x)