

# Servoy Application Server Upgrade

A Servoy Application Server installation can be upgraded from the command-line to new maintenance (x.x.x) or minor (x.x.0) version. Command-line upgrades to new major (x.0.0) versions are not supported, unless explicitly mentioned in the release notes.

In order to perform a command-line upgrade, the Servoy Application Server needs to be able to connect the internet and more explicitly to <http://www.servoy.com> for downloading the upgrade package.

## In This Chapter

- [Upgrading a Servoy Application Server](#)
- [Servoy Repository Upgrade](#)
- [Updating and Code Signing Heads-up](#)
- [Prevent Updating or Placement of Specific Files](#)
  - [Readonly files](#)
  - [Zero-size files](#)
- [Troubleshooting](#)
  - [Network issues](#)

## Upgrading a Servoy Application Server

To upgrade a Servoy Application Server the following steps must be followed:

1. Shutdown the Servoy Application Server
2. Open a command line/terminal window and browse to the `{servoyInstall}/application_server` directory
3. Execute one of the commands from the table below

	Command	Comment
Default	<code>java -jar servoy_updater.jar</code>	Upgrades to latest final release. Excludes alpha, beta and release candidates
Include beta versions	<code>java -jar servoy_updater.jar -beta</code>	Upgrades to latest non final release
Update to specific release (since Servoy 6.1)	<code>java -jar servoy_updater.jar -releaseNumber buildNumber</code>	Upgrades to a release based on the specific build number of the release. The build number for a specific release is mentioned in the <a href="#">announcement of the release</a> on the <a href="#">Servoy Talk forum</a> .

## Servoy Repository Upgrade

The update process will update only the Servoy Application Server installation. It does NOT update the [Servoy Repository](#) structure. When the upgraded Servoy Application Server is launched and it finds an incompatible version of the Servoy Repository, the launch will be aborted with the message that the Servoy Repository needs to be upgraded. See [Servoy Repository Upgrade](#) for more info on upgrading the Servoy Repository.

## Updating and Code Signing Heads-up

The code in the files updated by the updater will be properly signed by Servoy, similar to how all code of the Servoy Application Server is signed when first installed.

If the code signing of the installation has been replaced by custom signing, this means that part of the codebase is now signed using a custom certificate and part of the codebase is signed by Servoy's certificate. While this is not problematic from a technical perspective (the Smart Client will still work), it does mean that the user will be presented with multiple dialogs that asks if the user trusts the party that signed the code.

To resign the updated files with a custom certificate, the signing of all files that are part of the Servoy distribution needs to be checked and the files that are now signed using the Servoy certificate need to be resigned using the custom certificate.

For more on the topic of code signing see the [Code Signing](#) chapter.

---

## Prevent Updating or Placement of Specific Files

---

The update mechanism will automatically replace existing files if a new version is available in the update or will place a file that is part of the default distribution if it is missing in the existing installation. This might not be desired if for example certain plugins have been removed that are not used or different versions of libraries have been manually installed.

Both the update and the placement actions can be prevented, but understand that this means that any additions or fixes made to those files will not be available after the update. Worst-case scenario is that the old version is incompatible with the new Servoy version and things might not work as expected.

---

### Readonly files

---

The updater mechanism will not touch any file marked as read-only on the File system, so to prevent the updater to overwrite specific files, mark the files as read-only

---

### Zero-size files

---

By replacing a file that is shipped by default with an empty file that is read-only, the content of the file is effectively removed and as the file is read-only, the updater will not replace it with the default version

---

## Troubleshooting

---

---

### Network issues

---

Some Java environments start with IPv6 Stack. When actually running on IPv4, this might cause issues in the download of the update. Add `d-java.net.preferIPv4Stack=true` to the end of the update command.