

# Extensions

## Table of contents

1 Overview.....	2
2 Extension Installer.....	2
3 Project setup.....	3

## 1. Overview

Extensions add framework level functionality that is available to any other user of the framework. Extensions can include the following:

- Java class libraries which may include Ant tasks and types, and their own APIs.
- CTL command modules
- framework commands that live in `$CTL_HOME/bin`

The framework provides an extension installer that takes an extension archive and installs it into the framework.

## 2. Extension Installer

All extensions are installed via the `ctl-extension` command.

The `ctl-extension` command takes an extension JAR file, extracts it, reads the `MANIFEST.MF`, and copies files into the appropriate places under `$CTL_HOME` and `$CTL_BASE`.

The command usage of the command is shown below:

```
usage: ctl-extension [options]
options:
  -S,--nosetup          don't re-run setup
  -f,--file             file to install
  -h,--help            print this message
  -o,--overwrite       overwrite framework with files from extension
Examples:
ctl-extension -f extension-name.jar
```

Installed extensions can be found under `$CTL_HOME/lib/extensions` as described below:

```
ctl-home
|
|--bin                // shell scripts
|
|--classes           // bootstrap classes
|
|--depots            // project depots
|
|--lib
|   |--ant           // command modules
|   |--extensions  // framework extensions
```

```
+ -pkgs // ant home
```

### 3. Project setup

The `ctl-extension` command installs the modules into the framework but does not install them in the project depots. Use the `ctl-depot` command to pull those into the project depot:

```
ctl-depot -p project
```

For more information about the internals of an extension see the [reference](#).