

Packaging

Table of contents

1 Modules.....	2
2 Extensions.....	2
3 Seeds.....	2
4 Objects.....	3

Part of making your modules ready for broad use is packaging them so they can be distributed to your CTL installations. This document describes the options you have for packaging.

1. Modules

The most obvious and immediate way to distribute your module is via a module Jar file. The `build-type` command generates the module Jar file for you so you simply run that command:

```
ctl -m ProjectBuilder -c build-type -- -type YourModule
```

The built Jar file will reside under the directory specified by the `-targetdir` option. The module can then be deployed to a repository for distribution and then installed via the `Install-Module` wherever you want to install it.

```
ctl -m Managed-Entity -c Install-Module -- -module YourModule -version vers
```

Alternatively, if you want to distribute them more broadly, you can define your module as a `module-deployment` in the [deployments.properties](#) file and then run:

```
ctl-depot -p project -a install
```

You may need to distribute a collection of modules in one shot. The following two sections describe methods for distributing a single archive that contains multiple modules.

2. Extensions

An extension is an archive of modules and other supporting resources that effectively enhance the CTL framework. (See the [Extension](#) for more information about what can be included in the archive).

Building an extension archive is easy and accomplished via the `build-library` command with the `-archive extension` option:

```
ctl -m ProjectBuilder -c build-library -- -archive extension
```

The `build-library` command will generate a new Jar file that can be used by the `ctl-extension` command like so:

```
ctl-extension -f /path/to/targetdir/your-extension-vers.jar
```

Remember to run `ctl-depot` to pull those modules into your project depot:

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

3. Seeds

For ControlTier server users, especially those that use a sophisticated object model, a `seed` archive type is the preferred choice. A "seed" is short for "project seed", an archive of type

definitions that are loaded into the Workbench model for the specified project. When the seed is uploaded to a Workbench project all the modules are loaded and staged into its repository. They can be downloaded by CTL installations that use those modules.

To create a seed, run the `build-library` command with the `-archive seed` option:

```
ctl -m ProjectBuilder -c build-library -- -archive seed
```

The modules will be downloaded as needed by your objects or via the `ctl-depot` command like so:

```
ctl-depot -p project -a install
```

Note:

If your modules would be useful to others, consider posting an email to the ControlTier google group and propose it to be added as a Moduleforge project.

4. Objects

Some users prefer to distribute object data via package-centric release methodology rather than a client-server based approach. Object data can be maintained in an SCM repository and then packaged via the [ctl-archive](#) command.

To create an archive, run the `create` action from `ctl-archive`:

```
ctl-archive -a create -d src/myproject/datadir -f myproject-archive.jar
```

You can distribute this data using any preferred mechanism (eg, rpm). The post install procedure can then call `ctl-archive` to extract the data:

```
ctl-archive -a extract -p myproject -f myproject-archive.jar
```