

build-library

Command Reference

Table of contents

1 Description.....	2
2 Usage.....	2

1. Description

Build a type library.

Generates a library jar file from the types found in the `-basedir/modules`, saving the jar file into the `-targetdir`. The file created is named "library-`{opts.name}`.jar".

The builddir should contain types like this:

- `{opts.basedir}/modules/`
 - Type1/
 - type.xml
 - ...
 - Type2/
 - type.xml
 - ...

If the `-upload` flag is specified, the resulting library jar file will be uploaded to the server repository via the [load-library](#) command.

static: This command can be run outside of an object context.

2. Usage

```
ctl -m ProjectBuilder -c build-library [-archive <seed>]
[-basedir <>] [-createjar <true>] [-name <>] [-targetdir <>]
[-upload] [-version <1>]
```

2.1. Options

Option	Description	Type	Default
archive	<p><i>archive type to build. (seed or extension)</i></p> <p>There are two kinds of library archives.</p> <ul style="list-style-type: none"> • A "seed" archive can be uploaded to a ControlTier server and used to bootstrap projects in Workbench. (seed is default) 	string	seed

build-library

	<ul style="list-style-type: none"> An "extension" archive can be installed as a CTL extension. Extensions are useful for enhancing the CTL installation with new modules, executables and supporting Jar files (which may include Ant tasks and types). 		
basedir	<i>directory containing modules directory</i>	string	<code>\${entity.attribute.basedir}</code>
createjar	<i>if true, create the seed jar, otherwise only create the module jars</i>	string	true
name	<i>library name</i>	string	<code>\${context.name}</code>
targetdir	<i>directory to save library</i>	string	<code>\${entity.attribute.targetdir}</code>
upload	<i>upload library to repository</i>	boolean	
version	<i>archive version</i>	string	1