

## Edit Client Commands

The edit client commands are applied to different types of elements, such as *command*, *configuration*, *module*, *platform*, and *resource*. They provide the following operations:

1. *add* in order to add a new element in the DB;
2. *modify* in order to modify an element;
3. *prepare* in order to create either a template `.INI` file for a new element with default values or a `.INI` file for an existing element with DB values;
4. *remove* in order to remove an element in the DB.

The operation *clone* is only supported by the configuration command.

These commands can be used only after running `etics-get-project` and `etics-checkout`.

## Configure certificate

Several CLI commands require a valid user digital certificate. To install certificate, user should set parameters `x509_user_key` and `x509_user_cert` in `~/.etics.conf` so that they point to proper `*.pem` files.

If you only have a p12 format certificate, you'll find a recipe here: [HowToConvertCertificatesFromP12ToPem](#)

## etics-platform

`etics-platform` is used for the element *platform*. It can be used as follows:

```
> etics-platform [options] <platform-name _moz-userdefined=""></platform-name>
```

options are:

```
-f, --force
-i, --input
-o, --output
--param
```

To create a `.INI` file we can do as follows:

```
> etics-platform prepare <platform-name _moz-userdefined=""></platform-name>
```

The information related with the element *platform* will be copied in the default file called `platform-<platform-name _moz-userdefined="">.INI</platform-name>`. If the file already exists, the command returns an error.

To add a new *platform* in the DB we can do as follows:

```
> etics-platform add <platform-name _moz-userdefined=""></platform-name>
```

The information related with the element *platform* will be provide by the user interactively.

```
> etics-platform add -i <platform-name _moz-userdefined=""></platform-name>
```

The information related with the element *platform* will be provide by the input file.

```
> etics-platform add --param "displayName=paperino, descritpion='new platform for'"
<platform-name _moz-userdefined=""></platform-name>
```

The information related with the element *platform* will be provide by using the option `param`.

To modify an existing *platform* we can do as for the operation add.

## etics-resource

`etics-resource` is used for the element *resource*. It can be used as follows:

```
> etics-platform [options] <resource-name _moz-userdefined=""></resource-name>
```

options are:

```
-f, --force  
-i, --input  
-o, --output  
--param
```

To create a `.INI` file we can do as follows:

```
> etics-resource prepare <resource-name _moz-userdefined=""></resource-name>
```

The information related with the element *resource* will be copied in the default file called `resource-<resource-name _moz-userdefined="">.INI</resource-name>`. If the file already exists, the command returns an error.

To add a new *resource* in the DB we can do as follows:

```
> etics-resource add <resource-name _moz-userdefined=""></resource-name>
```

The information related with the element *resource* will be provide by the user interactively.

```
> etics-resource add -i <resource-name _moz-userdefined=""></resource-name>
```

The information related with the element *resource* will be provide by the input file.

```
> etics-resource add --param "displayName=dune, description='new ETICS resource'"  
<resource-name _moz-userdefined=""></resource-name>
```

The information related with the element *resource* will be provide by using the option `param`.

To modify an existing *resource* we can do as for the operation add.

## etics-module

`etics-module` is used for the element *component* and *subsystem*. It can be used as follows:

```
> etics-module [options] <module-name _moz-userdefined=""></module-name>
```

options are:

```
-f, --force  
-i, --input  
-o, --output  
--parent  
--component  
-s, --subsystem  
--param
```

If the option `parent` is not used, we assume that it is the project. It is mandatory when the *module* is a *component* of a *subsystem*.

To create a `.INI` file we can do as follows:

```
> etics-module prepare --component/--subsystem <module-name
_moz-userdefined=""></module-name>
```

The information related with the element *module* will be copied in the default file called `{component/subsystem}-<module-name _moz-userdefined="">.INI</module-name>`. If the file already exists, the command returns an error.

To add a new *module* in the DB we can do as follows:

```
> etics-module add --component/--subsystem <module-name _moz-userdefined=""></module-name>
```

The information related with the element *module* will be provide by the user interactively.

```
> etics-module add -i --component/--subsystem <module-name
_moz-userdefined=""></module-name>
```

The information related with the element *module* will be provide by the input file.

```
> etics-module add --param "displayName=mo, descriptpion='new ETICS m'" <module-name
_moz-userdefined=""></module-name>
```

The information related with the element *module* will be provide by using the option `param`.

To modify an existing *module* we can do as for the operation `add`.

## etics-command

`etics-command` is used for the element *BuildCommand*, *VcsCommand*, and *TestCommand*. It can be used as follows:

```
> etics-command [options] <command-name _moz-userdefined=""></command-name>
```

options are:

```
-f, --force
-i, --input
-o, --output
--vcsc
--buildc
--testc
--param
```

The options `vcsc`, `buildc`, and `testc` are mandatory.

To create a `.INI` file we can do as follows:

```
> etics-command prepare --vcsc/--buildc/--testc <command-name
_moz-userdefined=""></command-name>
```

The information related with the element *command* will be copied in the default file called `{BuildCommand/TestCommand/VcsCommand}-<command-name`

`_moz-userdefined="">.INI</command-name>`. If the file already exists, the command returns an error.

To add a new *command* in the DB we can do as follows:

```
> etics-command add --vcsc/--buildc/--testc <command-name  
_moz-userdefined=""></command-name>
```

The information related with the element *command* will be provide by the user interactively.

```
> etics-command add -i --vcsc/--buildc/--testc <command-name  
_moz-userdefined=""></command-name>
```

The information related with the element *command* will be provide by the input file.

```
> etics-command add --param "displayName=test, descriptpion='new ETICS test command'"  
--testc <command-name _moz-userdefined=""></command-name>
```

The information related with the element *command* will be provide by using the option param.

To modify an existing *command* we can do as for the operation add.

## etics-configuration

`etics-configuration` is used for the element *configuration*. It can be used as follows:

```
> etics-configuration [options] <configuration-name  
_moz-userdefined=""></configuration-name>
```

options are:

```
-f, --force  
-i, --input  
-o, --output  
-m, --module  
--param
```

To prepare the INI file you can run the commad:

```
> etics-configuration prepare -m <module-name _moz-userdefined=""> <configuration-name  
_moz-userdefined=""></configuration-name></module-name>
```

**In the case the** `<configuration-name _moz-userdefined=""></configuration-name>` **is** `org.etics.build-system.java-util.HEAD` **and the** `<module-name _moz-userdefined=""></module-name>` **is** `org.etics.build-system.java-util`, **the INI file is the following:**

```
[Configuration-org.etics.build-system.java-utils.HEAD]  
majorversion = 0  
modulename = org.etics.build-system.java-utils  
displayname = etics Java Utils v. 0.0.1  
description = etics Java Utils v. 0.0.1  
repository = None  
projectname = org.etics  
packagename = build-system-java-utils  
revisionversion = 0  
vcsroot = :pserver:anonymous@etics.cvs.cern.ch:/cvs/etics  
testprofile = None  
licencetype = etics  
tag = HEAD
```

## ClientEditMode < ETICS < TWiki

```
moduledescription = ETICS Java utils
path = None
moduletype = C
buildprofile = None
age = 1
minorversion = 0
```

```
[Configuration-org.etics.build-system.java-utils.HEAD:Platform-default]
vcscommand = CVS
testcommand = None
buildcommand = ETICS Web Service build command
```

```
[Configuration-org.etics.build-system.java-utils.HEAD:Platform-default:Environment]
var1 = None
```

```
[Configuration-org.etics.build-system.java-utils.HEAD:Platform-default:Property]
var1 = None
```

```
[Configuration-org.etics.build-system.java-utils.HEAD:Platform-default:Dependency]
moduleconf1 = ant:ant v. 1.6.2
moduleconf0 = jdk:jdk v. 1.5.0_06
moduleconf2 = mysql-jdbc:mysql-jdbc v. 3.1.10
```

---

This topic: [ETICS > ClientEditMode](#)

Topic revision: r6 - 2007-01-10 - MebSter



Copyright &© 2008-2022 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.

or Ideas, requests, problems regarding TWiki? use [Discourse](#) or [Send feedback](#)