

EOF-Using EOF-EOF Best Practices

Overview

First, do no harm - Read and obey [The EOF Commandments](#) before reading this.

Creating EOEnterpriseObjects

Rather than use

```
MyEO eo = new MyEO();
ec.insertObject(eo);
```

Prefer this form:

```
MyEO eo = (MyEO) EOUtilities.createAndInsertInstance(ec, "MyEO");
```

This has two main advantages. One, it prevents the use of an EO which has not been inserted into an EOEditingContext (see [The EOF Commandments](#)). Two, it supports the case where two or more entities in the EOModel(s) are implemented by the same Java class.

A reasonable question to ask here is, If this is good:

```
MyNewPage nextPage = (MyNewPage)pageWithName(MyNewPage.class.getName());
```

then why not this?

```
MyEO eo = (MyEO) EOUtilities.createAndInsertInstance(ec, MyEO.class.getName());
```

The key here is that the second parameter to `createAndInsertInstance` is the EOEntity name from the model, not the class name. For component creation it is the class name. The thing is that the entity name and the class name are often not the same. The class names will include package name, but the entity names will not. Also, I've run across cases where multiple entities in the model are implemented by the same class. Gary Teter makes good use of this in the WireHose frameworks.