

# Using a custom EOEditingContext (ERXEC) Subclass

## Create the EditingContext Subclass

```
public class MyEditingContext extends ERXEC {

    public MyEditingContext(EOObjectStore anObjectStore) {
        super(anObjectStore);
    }

    public MyEditingContext() {
        super();
    }

    // Your custom logic here...

}
```

## Create the ERXEC.DefaultFactory Subclass

```
public class MyEditingContextFactory extends ERXEC.DefaultFactory {

    public MyEditingContextFactory() {
        super();
    }

    protected EOEditingContext _createEditingContext(EOObjectStore parent) {
        return new MyEditingContext(parent == null ? EOEditingContext.defaultParentObjectStore() : parent);
    }

}
```

## Tell Wonder to Use Your Custom Factory

In your Application constructor, set your custom editing context factory as the default factory for ERXEC as shown here for example:

```
public Application() {
    super();

    // Configure the Editing Context factory for my subclass of ERXEC
    ERXEC.setFactory( new MyEditingContextFactory() );

    // More app initialization code.....

}
```

## Creating an Instance of Your Custom Editing Context Subclass

```
EOEditingContext ec = ERXEC.newEditingContext();
```