Java

Apache Cayenne Intro

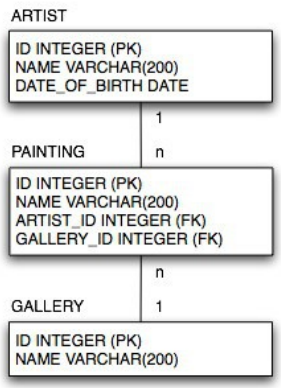
Ilya Drabenia
Software Engineer,
ObjectStyle LLC

objectstyle
/*open_software_engineering*/

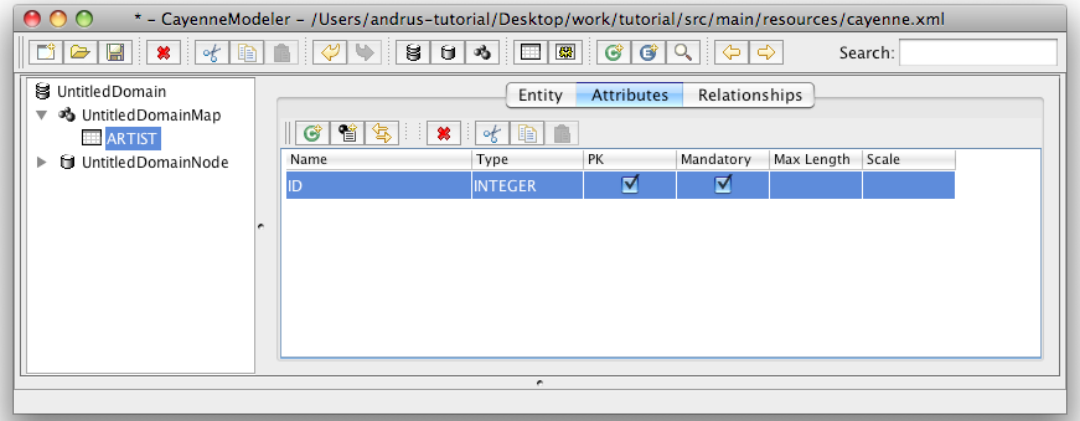
What is Cayenne?

- GUI tool for design entities and schema
- Code First and Schema First mapping
- Good integration with ant and maven
- Great support of unit and integration testing
- Support of multi-tenancy and SaaS
- Support of Dependency Injection

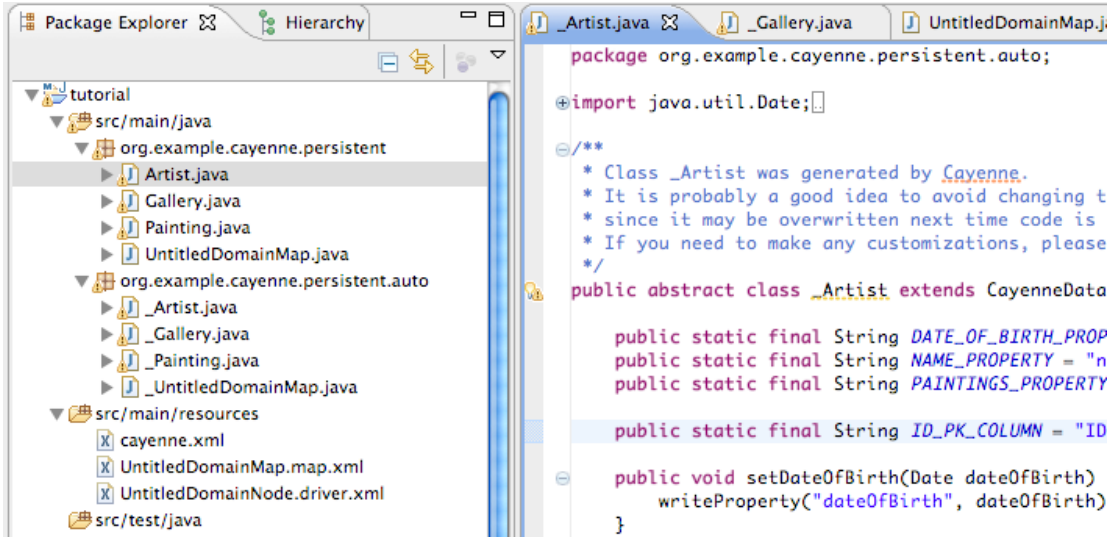
Five minutes tutorial



Design



Generate



Five minutes tutorial

That’s all! Model ready to use!

Five minutes tutorial

Main class

```
package org.example.cayenne;

import org.apache.cayenne.ObjectContext;
import org.apache.cayenne.access.DataContext;

public class Main {

    public static void main(String[] args) {
        ObjectContext context = DataContext.createDataContext();
    }

}
```

APACHE CAYENNE

Five minutes tutorial

Let’s create new entities

```
Gallery metropolitan =
context.newObject(Gallery.class);
metropolitan.setName("Metropolitan Museum of
Art");

Painting girl = context.newObject(Painting.class);
girl.setName("Girl Reading at a Table");
```

Specify relationships

```
picasso.addToPaintings(girl);
picasso.addToPaintings(stein);

girl.setGallery(metropolitan);
stein.setGallery(metropolitan);
```

And commit changes

```
context.commitChanges();
```

APACHE CAYENNE

Five minutes tutorial

Let’s query Picasso entity

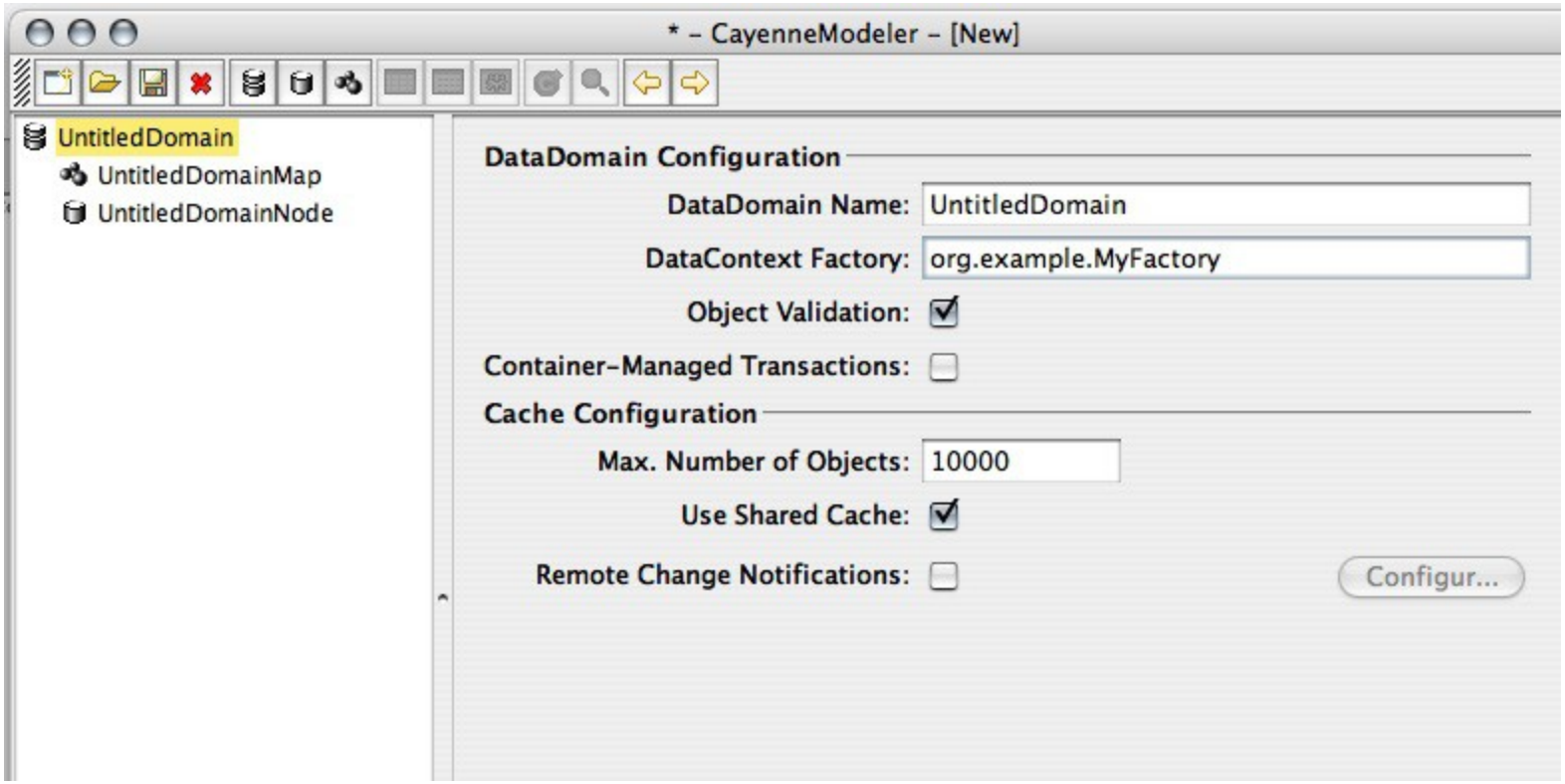
```
Expression qualifier = ExpressionFactory.matchExp(Artist.NAME_PROPERTY, "Pablo Picasso");
SelectQuery select = new SelectQuery(Artist.class, qualifier);
Artist picasso = (Artist) DataObjectUtils.objectForQuery(context, select);
```

And delete it

```
if (picasso != null) {
    context.deleteObject(picasso);
    context.commitChanges();
}
```

APACHE CAYENNE

Cayenne Modeler



Cayenne Modeler

Main features:

1. Configuration of entities to tables mapping
2. Generation of xml files with mapping
3. Generation of Java classes
4. Project creation using existing DB
5. Database schema migration

Generated Entity

```
public abstract class _RegistrationInfo extends
CayenneDataObject {

    public static final String EMAIL_PROPERTY = "email";
    public static final String PASSWORD_PROPERTY =
"password";
    public static final String VERSION_PROPERTY =
"version";
    public static final String ID_PK_COLUMN = "id";

    public void setEmail(String email) {
        writeProperty("email", email);
    }
    public String getEmail() {
        return (String)readProperty("email");
    }

    public void setPassword(String password) {
        writeProperty("password", password);
    }
    public String getPassword() {
        return (String)readProperty("password");
    }

}
```

APACHE CAYENNE

Data Context

Major methods of DataContext:

- public <T> T newObject(Class<T> persistentClass)
- public void deleteObject(Object object)
- public java.util.List performQuery(Query query)
- public void commitChanges()
- public void rollbackChanges()

Advanced Features

- First level cache and query cache
- Remote Object Persistence support
- Lifecycle callbacks
- Lazy and eager loading
- Batch query support
- Advanced transaction management

APACHE CAYENNE

Why Cayenne?

- Simple to learn and start
- Rapid Application Development support
- Support of all popular RDBMS
- High quality and great performance
- Cayenne do not spent much time on app start
- More than 1 000 projects in

APACHE CAYENNE

Thanks!

Questions?