

# OJB - Write Tests

by Armin Waibel

## Table of contents

1 Introduction.....	2
2 How to write a new Test.....	2
2.1 The Test Class.....	2
2.2 Persistent Objects used by Test.....	3
2.3 Test Class Metadata.....	3

## 1. Introduction

As described in [test suite section](http://../docu/testing/testsuite.html) (../docu/testing/testsuite.html) OJB emphasise quality assurance and provide a huge test suite. But it is impossible to cover all parts of OJB with tests and OJB will never be perfect (of course it's nearly perfect ;-)), thus if you miss a test or found an bug don't hesitate, write your own test and send it to [the lists](http://../mail-lists.html) (../mail-lists.html) or attach it in the [bug report](http://issues.apache.org/jira/secure/BrowseProject.jspa?id=10700) (http://issues.apache.org/jira/secure/BrowseProject.jspa?id=10700)

## 2. How to write a new Test

Before start writing your own test case please pay attention of these rules.

### 2.1. The Test Class

All test classes have to inherit from `org.apache.ojb.junit.OJBTestCase` and have to provide a static main method to start the Junit test:

```
public class MyTest extends OJBTestCase
{
    public static void main(String[] args)
    {
        String[] arr = {MyTest.class.getName()};
        junit.textui.TestRunner.main(arr);
    }

    public void testMyFirstOne()
    {
        ....
    }
}
```

In package `org.apache.ojb.junit` can be found some test classes for specific circumstances:

- `org.apache.ojb.junit.PBTestCase` - Provide a public `org.apache.ojb.broker.PersistenceBroker` field.
- `org.apache.ojb.junit.ODMGTestCase` - Provide public `org.odmg.Implementation` and `org.odmg.Database` fields.
- `org.apache.ojb.junit.JUnitExtensions` - Provide base classes for write multithreaded test classes. More info see javadoc comment of this class.

A test case for the PB-API may look like:

```
public class ReferenceRuntimeSettingTest extends PBTestCase
{
    public static void main(String[] args)
```

## OJB - Write Tests

```
{
    String[] arr = {ReferenceRuntimeSettingTest.class.getName()};
    junit.textui.TestRunner.main(arr);
}

public void testChangeReferenceSetting()
{
    ClassDescriptor cld = broker.getClassDescriptor(MainObject.class);
    // and so on
    ....
}
```

The PersistenceBroker cleanup is done by *PBTestCase*.

### 2.2. Persistent Objects used by Test

We recommend to introduce separate persistent objects for each TestCase class. In [test suite](#) ([../docu/testing/testsuite.html](#)) two concepts are used:

- Include your persistent objects as *public static classes* in your test class.
- Separate your test class in an independent package and include the test case and all persistent object classes in this new package.

### 2.3. Test Class Metadata

Currently all test specific object metadata (class-descriptor used for tests) are shared among several xml files. The naming convention is `repository_junit_XXX.xml`. Thus metadata for new tests should be included in one of the existing junit repository (sub) files or written in an new separate one and included in repository main file.

```
<!DOCTYPE descriptor-repository PUBLIC
    "-//Apache Software Foundation//DTD OJB Repository//EN"
    "repository.dtd"
[
<!ENTITY database SYSTEM "repository_database.xml">
<!ENTITY internal SYSTEM "repository_internal.xml">
<!ENTITY user SYSTEM "repository_user.xml">

<!-- here the junit include files begin -->
<!ENTITY junit SYSTEM "repository_junit.xml">
<!ENTITY junit_odmg SYSTEM "repository_junit_odmg.xml">
<!ENTITY junit_otm SYSTEM "repository_junit_otm.xml">
<!ENTITY junit_ref SYSTEM "repository_junit_reference.xml">
<!ENTITY junit_meta_seq SYSTEM "repository_junit_meta_seq.xml">
<!ENTITY junit_model SYSTEM "repository_junit_model.xml">
<!ENTITY junit_cloneable SYSTEM "repository_junit_cloneable.xml">

<!ENTITY junit_myfirsttest SYSTEM "repository_junit_myfirsttest.xml">
]>
<descriptor-repository version="1.0" isolation-level="read-uncommitted"
```

```
    proxy-prefetching-limit="50">

    <!-- include all used database connections -->
    &database;

    <!-- include ojb internal mappings here -->
    &internal;

    <!-- include user defined mappings here -->
    &user;

    <!-- include mappings for JUnit tests -->
    <!-- This could be removed (with <!ENTITY entry),
         if junit test suite was not used
    -->
    &junit;
    &junit_odmg;
    &junit_otm;
    &junit_ref;
    &junit_meta_seq;
    &junit_model;
    &junit_cloneable;

    &junit_myfirsttest;
```