

**Computer Science and Engineering**  
**Project of Software Engineering 2**

**Inspection Document**

**Authors:**

Alberto Zeni 813977

Manuel Parenti 876085

**Reference Professor:** Mottola Luca

Academic Year 2016–2017

**Table of Contents**

[**1 Classes that were assigned to the group**: 2](#_Toc473987296)

[**2 Functional role of assigned set of classes** 2](#_Toc473987297)

[**3 List of issues found by applying the checklist** 3](#_Toc473987298)

[3.1 Naming conventions issues 3](#_Toc473987299)

[3.2 Indentation issues 3](#_Toc473987300)

[3.3 Braces issues 3](#_Toc473987301)

[3.4 File organization issues 4](#_Toc473987302)

[3.5 Wrapping lines issues 4](#_Toc473987303)

[3.6 Comments issues 4](#_Toc473987304)

[3.7 Java source files issues 5](#_Toc473987305)

[3.8 Package and Import Statements issues 5](#_Toc473987306)

[3.9 Class and interface declarations issues 5](#_Toc473987307)

[3.10 Initializations and declarations issues 5](#_Toc473987308)

[3.11 Method calls issues 5](#_Toc473987309)

[3.12 Arrays issues: 5](#_Toc473987310)

[3.13 Object comparison issues 6](#_Toc473987311)

[3.14 Output format issues 6](#_Toc473987312)

[3.15 Computation, comparisons and assignments issues: 7](#_Toc473987313)

[3.16 Exceptions issues 7](#_Toc473987314)

[3.17 Flow of control issues 7](#_Toc473987315)

[3.18 Files issues 7](#_Toc473987316)

[**4 Additional Problems** 7](#_Toc473987317)

[**5 Additional Information** 8](#_Toc473987318)

[5.1 Hours of Work 8](#_Toc473987319)

[5.2 Used Tools 8](#_Toc473987320)

**1 Classes that were assigned to the group:**

The class that was assigned to our group is: ServiceXaWrapper.java  
And its location in the ofbiz directory is the following:

../apache-ofbiz-16.11.01/framework/service/src/main/java/org/apache/ofbiz/service/ServiceXaWrapper.java

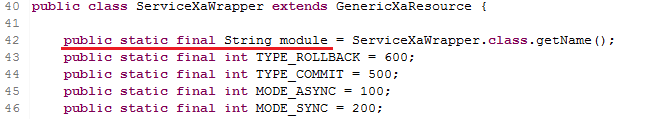
# **2 Functional role of assigned set of classes**

The class we need to analyze is a wrapper for a Xa Resource. This means it acts like a translator for the instructions it receives in order to make them comprehensible for the Xa Resource. In particular, our class focus on running services on commit and rollback methods, defined in the generic class that our class extends.

# **3 List of issues found by applying the checklist**

## **3.1 Naming conventions issues**

The constant “module” is not written in capital letters.

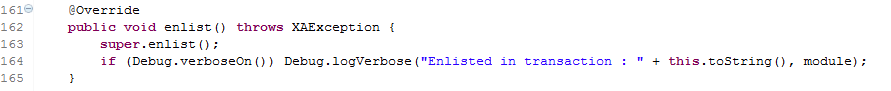


## **3.2 Indentation issues**

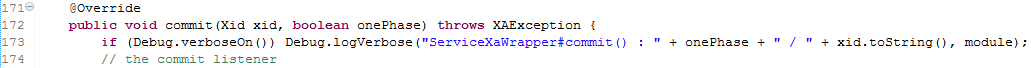
No indentation issues have been found.

## **3.3 Braces issues**

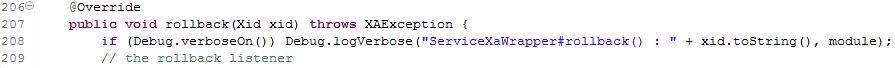
The statement following the “if” clause at line 164 is not included in curly braces. Also, it could become a shorter and more readable line.



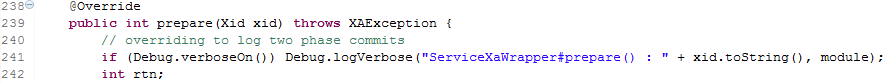
The statement following the “if” clause at line 173 is not included in curly braces. Also, it could become a shorter and more readable line.



The statement following the “if” clause at line 208 is not included in curly braces. Also, it could become a shorter and more readable line.



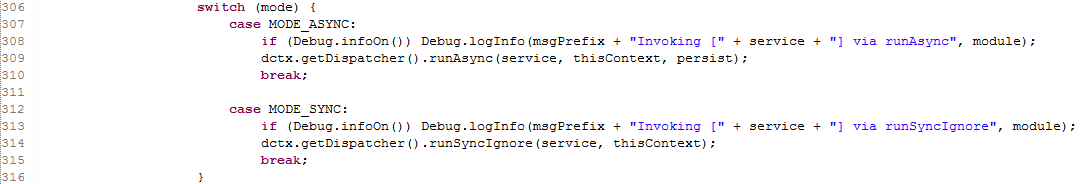
The statement following the “if” clause at line 241 is not included in curly braces. Also, it could become a shorter and more readable line.



The statement following the “if” clause at line 249 is not included in curly braces. Also, it could become a shorter and more readable line.



The statements following the “if” clauses at lines 308 and 313 are not included in curly braces. Also, they could become shorter and more readable lines.



The statements following the “if” clause at line 350 is not included in curly braces. Also, it could become a shorter and more readable line.



Every other line of code follows the Kernighan & Ritchie style.

## **3.4 File organization issues**

Line 79 is longer than 120 characters (126).



Line 90 is longer than 120 characters (144).



Line 128 is longer than 120 characters (128).



Line 139 is longer than 120 characters (146).



Line 255 is longer than 120 characters (149).



## **3.5 Wrapping lines issues**

No wrapping lines issues have been found.

## **3.6 Comments issues**

No comments issues have been found.

## **3.7 Java source files issues**

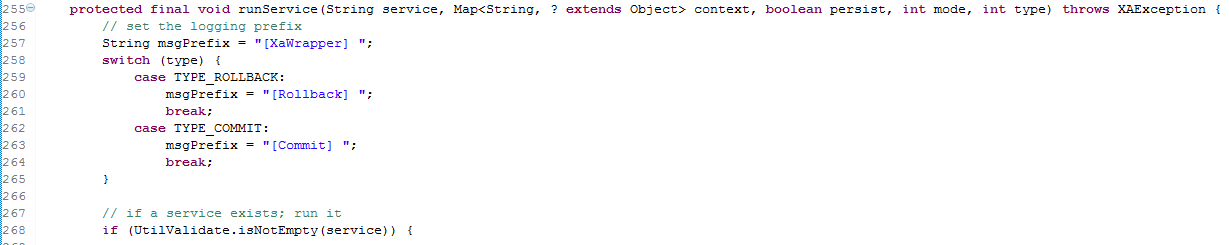
No java source file issues have been found.

## **3.8 Package and Import Statements issues**

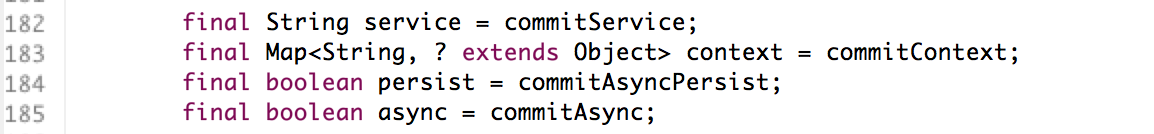
No issues concerning package and import statements have been found.

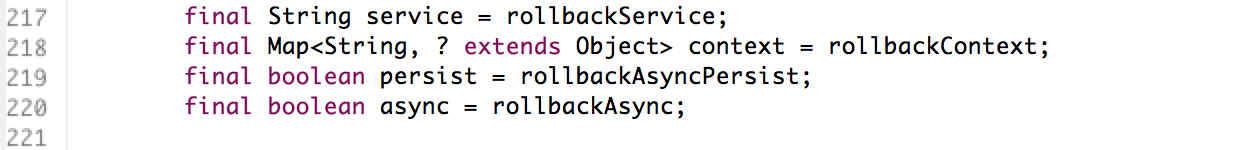
## **3.9 Class and interface declarations issues**

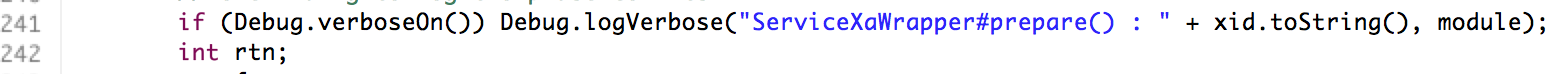
The method runService defined at line 255 is 100 lines long. We think that it should be considerably shorter, this issue might be fixed dividing this method in at least 2-3 submethods.



## **3.10 Initializations and declarations issues**

The variables ‘service’, ‘context’, ‘persist’, ‘async’ declared starting from line 182 are not declared at the beginning of the block.

The variables ‘service’, ‘context’, ‘persist’, ‘async’ declared starting from line 217 are not declared at the beginning of the block.

The variable ‘rtn’ declared at line 242 is not declared at the beginning of the block.

## **3.11 Method calls issues**

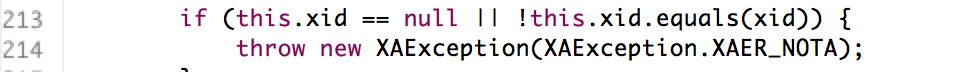
No method calls issues have been found.

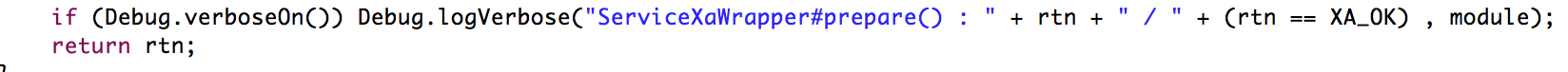
## **3.12 Arrays issues:**

No array issues have been found.

## **3.13 Object comparison issues**

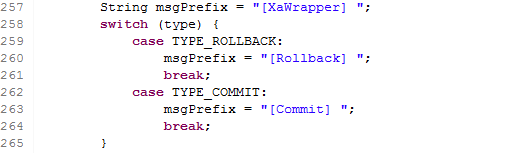
The xid variable at line 178 is compared using the ‘==’ instead of equal.

The xid variable at line 213 is compared using the ‘==’ instead of equal.

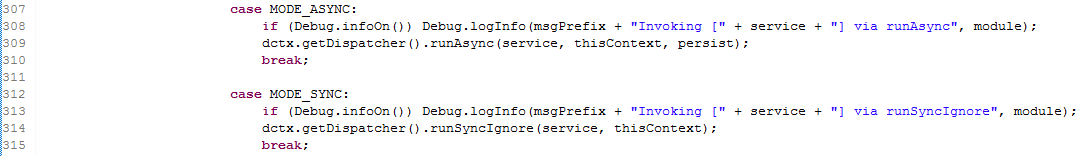
The rtn variable at line 249 is compared using the ‘==’ instead of equal.

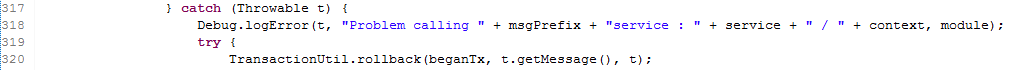
## **3.14 Output format issues**

In terms of output readability there are no issues, but we think that the way in which the strings are dealt with in the following snapshots of code could be changed in a better implementation. The string we are referring to is “msgPrefix” and it is defined at line 257.



It is used in the following outputs:







We think that a more general and safe way to write these outputs is to write a space after the “msgPrefix” variable and to delete the spaces in the “msgPrefix” text.

## **3.15 Computation, comparisons and assignments issues:**

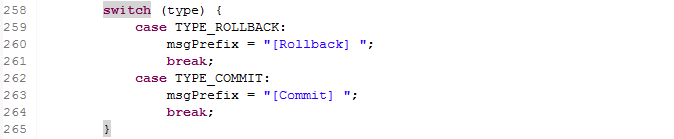
No computation, comparisons or assignments issues have been found.

## **3.16 Exceptions issues**

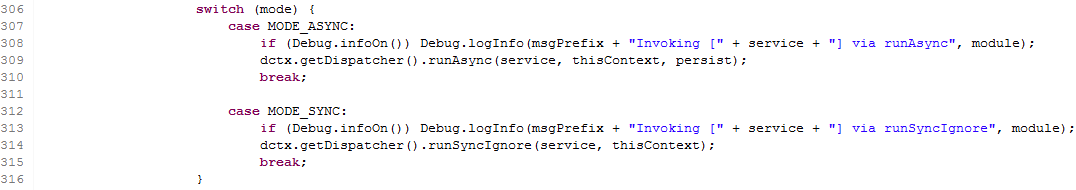
No exceptions issues have been found.

## **3.17 Flow of control issues**

The switch at line 258 hasn’t a default branch.



The switch at line 306 hasn’t a default branch.



## **3.18 Files issues**

No files are opened by the methods of the class, so no issues have been found.

# **4 Additional Problems**

Many references about the xid and the active objects, that are presumably part of the GenericXaResource class, are in the code, but since this is an extended class we don’t know what they are. Because of this they could be sources for problems during the implementation of our class.

# **5 Additional Information**

## **5.1 Hours of Work**

Time spent to write this document:

Manuel Parenti: 7 hours

Alberto Zeni: 7 hours

## **5.2 Used Tools**

The tools that supported us are:

MS Office Word 2016: to write and redact this document;

Eclipse: to read the java code of the class.