



## OFBiz - Code Inspection

February 5, 2017

A.Y. 2016/2017

Bolshakova Liubov, matr. 876911

Gao Xiao, matr. 876265

Kang Shuwen, matr. 876245

# Contents

|          |  |           |
|----------|--|-----------|
| <b>1</b> | <b>CODE DESCRIPTION</b>                      | <b>2</b>  |
| 1.1      | Assigned Classes . . . . .                   | 2         |
| 1.1.1    | RequirementServices.java . . . . .           | 2         |
| 1.1.2    | EntityEcaCondition.java . . . . .            | 2         |
| 1.2      | Introduction . . . . .                       | 2         |
| 1.2.1    | Apache OFBiz . . . . .                       | 2         |
| 1.2.2    | Order Manager . . . . .                      | 2         |
| 1.3      | Functional Role of Classes . . . . .         | 3         |
| 1.3.1    | RequirementServices.java . . . . .           | 3         |
| 1.3.2    | EntityEcaCondition.java . . . . .            | 5         |
| <b>2</b> | <b>INSPECTION RESULTS</b>                    | <b>7</b>  |
| 2.1      | Issues of RequirementServices.java . . . . . | 7         |
| 2.1.1    | Naming Conventions . . . . .                 | 7         |
| 2.1.2    | Indention . . . . .                          | 7         |
| 2.1.3    | Braces . . . . .                             | 7         |
| 2.1.4    | File Organization . . . . .                  | 7         |
| 2.1.5    | Wrapping Lines . . . . .                     | 8         |
| 2.1.6    | Java Source Files . . . . .                  | 8         |
| 2.1.7    | Output Format . . . . .                      | 8         |
| 2.2      | Issues of EntityEcaCondition.java . . . . .  | 8         |
| 2.2.1    | Indention . . . . .                          | 8         |
| 2.2.2    | Braces . . . . .                             | 9         |
| 2.2.3    | File Organization . . . . .                  | 9         |
| 2.2.4    | Wrapping Lines . . . . .                     | 9         |
| 2.2.5    | Java Source Files . . . . .                  | 9         |
| <b>3</b> | <b>APPENDIX</b>                              | <b>10</b> |
| 3.1      | Software and Tools Used . . . . .            | 10        |
| 3.2      | Hours of Work . . . . .                      | 10        |
| <b>4</b> | <b>REFERENCES</b>                            | <b>11</b> |

# 1 CODE DESCRIPTION

## 1.1 Assigned Classes

The assigned classes of our group are listed as follows:

### 1.1.1 RequirementServices.java

The class is located in: `../apache-ofbiz-16.11.01/applications/order/src/main/java/org/apache/ofbiz/order/requirement/RequirementServices.java`

### 1.1.2 EntityEcaCondition.java

The class is located in: `../apache-ofbiz-16.11.01/framework/entityext/src/main/java/org/apache/ofbiz/entityext/eca/EntityEcaCondition.java`

## 1.2 Introduction

### 1.2.1 Apache OFBiz

Apache OFBiz is an open source enterprise resource planning (ERP) system. It provides a suite of enterprise applications that integrate and automate many of the business processes of an enterprise.

All of Apache OFBiz functionality is built on a common framework. The functionality can be divided into the following distinct layers:

- **Presentation Layer**

Apache OFBiz uses the concept of "screens" to represent the Apache OFBiz pages.

- **Business Layer**

The business, or application layer defines services provided to the user.

- **Data Layer**

The data layer is responsible for database access, storage and providing a common data interface to the Business layer.

### 1.2.2 Order Manager

Order Manager is an application inside the OFBiz project, which is responded to due with the order processing. The Order Manager contains server sub-services, including:

- Main
- Requests
- Quotes

- Order List
- Find Orders
- Order Entry
- Returns
- Requirements
- Reports
- Stats

## 1.3 Functional Role of Classes

### 1.3.1 RequirementServices.java

One of our assigned classes, RequirementServices.java located in the business layer, is designed to provide the requirement service in the Order Manager application. As shown in Figures below, the requirement service provides four screens to search for requirements, which are

- **REQUIREMENTS**
- **APPROVE REQUIREMENTS**
- **FIND APPROVED REQUIREMENTS BY SUPPLIER**
- **APPROVED PRODUCT REQUIREMENTS.**

The requirement service also provides a screen named **New Requirement**, in order to manually add new requirement as well as edit existing requirement.

Figure 1: REQUIREMENTS

The screenshot shows the 'Find Not Approved Requirements' search form in the OFBiz application. The form includes search options for Requirement Id, Requirement Type Id, Facility Id, Product Id, Description, Requirement Start Date, and Required By Date. It features a 'Find' button and a 'Submit' button. The search results table has columns: REQUIREMENT ID, REQUIREMENT TYPE ID, FACILITY ID, PRODUCT ID, REQUIREMENT START DATE, REQUIRED BY DATE, QUANTITY, and SELECT.

Figure 2: APPROVE REQUIREMENTS

The screenshot shows the 'Find Approved Requirements By Supplier' search form in the OFBiz application. The form includes search options for Party Id, Supplier Currency Uom Id, # Products, Bill To Customer Party Id, Facility Id, and Prepare Order. It features a 'Find' button and a 'Submit' button. The search results table has columns: PARTY ID, SUPPLIER CURRENCY UOM ID, # PRODUCTS, BILL TO CUSTOMER PARTY ID, FACILITY ID, and PREPARE ORDER.

Figure 3: FIND APPROVED REQUIREMENTS BY SUPPLIER

The screenshot shows the 'Find Approved Product Requirements' search form in the OFBiz application. The form includes search options for Requirement Id, Bill To Customer Party Id, Supplier, unassigned Requirements, Facility Id, Product Id, and requirement By Date. It features a 'Find' button and a 'Print (PDF)' button. The search results table has columns: REQUIREMENT ID, PRODUCT ID, INTERNAL NAME, FACILITY ID, SUPPLIER, SUPPLIER PRODUCT ID, UPCA, MIN ORDER QTY, LAST PRICE, REQUIRED BY DATE, QUANTITY, and COMMENTS.

Figure 4: APPROVED PRODUCT REQUIREMENTS

To achieve the business logic of functions displayed above, there are

basically four methods in this class, and the particular functional role of each method is described in detail in the following paragraphs.

- **getRequirementsForSupplier()**

This method preforms the most important role in the requirement service, by reading the searching condition, provided by the user, from the screen, and combining the condition into query, and finally return back the result from the database as a map to support the information presentation on the screen. Basically, this method provides the requirement searching result by either getting the requirements for a given supplier, unassigned requirements, or requirements for all suppliers.

- **createAutoRequirementsForOrder()**

This method is designed to automatically create a requirement for an order, once a sales order status changes from CREATE to APPROVED.

- **createATPRequirementsForOrder()**

The strategy in this service is to begin making requirements when the product falls below the `ProductFacility.minimumStock`. Because the `minimumStock` is an upper bound, the quantity to be required is either that required to bring the ATP(Available to Promise) back up to the `minimumStock` level or the amount ordered, whichever is less.

If there is a way to support `reorderQuantity` without losing the order item to requirement association data, then this service should be updated.

The result is that this service generates many small requirements when stock levels are low for a product, which is perfectly fine since the system is capable of creating POs in bulk from aggregate requirements. The only concern would be a UI to manage numerous requirements with ease, preferably by aggregating on `productId`.

- **updateRequirementsToOrdered()**

This method is designed to change status of requirement to ORDERED.

### 1.3.2 EntityEcaCondition.java

There are basically six methods contained in this class, which are further discussed as follows.

- **eval()**

This method is for check if the dispatch messages are sent out successfully , if it is successes,return right, else return false. This is the

whole process of check a message. At first the method get the information about each dispatch, and check the basic information: if the order number, dctx and value exists, then get the satisfied name of dispatch, evaluate the condition and invoke the actions, then check if information in to message is in a specific format, finally check if any messages were returned send them out, only if all of them are correct, this dispatch process can be seen as successes.

- **getLValue()**

This method is used by the eval() method, it returns the specific lhsValueName of the dispatch information.

- **getRValue()**

This method is used by the eval() method, it returns the specific rhsValueName of the dispatch information.

- **getOperator()**

This method is used by the eval() method, it returns the specific operator of the dispatch information.

- **toString()**

This method is used by the eval() method for generating messages, it edits the information to the message and returns the message as a complete string.

- **getFieldNames()**

This method is for generating a list of lhsValueName and rhsValueName which exist in the dispatch information and return this list as fieldNameList.

## 2 INSPECTION RESULTS

In this section, we present all the issues that we found during code inspection, according to the checklist provided by the Assignment. Notice that, the notation Point related to the check point numbered in the code inspection assignment document.

### 2.1 Issues of RequirementServices.java

#### 2.1.1 Naming Conventions

- **Point 7**

Two constant attributes of the class do not follow the naming convention for constants.

- module should be MODULE
- resource\_error should be RESOURCE\_ERROR

#### 2.1.2 Indention

- **Point 9**

Tabs are used to indent.

#### 2.1.3 Braces

- **Point 11**

For the following lines, the single statement to execute is not surrounded by curly braces.

- Line 133
- Line 173

#### 2.1.4 File Organization

- **Point 13**

Many lines of code are not broken up properly and exceed 80 characters.

- **Point 14**

For the following lines, the line length exceed 120 characters.

- Line 130
- Line 162
- Line 225



- Line 230
- Line 285
- Line 292
- Line 318
- Line 320

### **2.1.5 Wrapping Lines**

- **Point 16**

Higher-level breaks are not used.

### **2.1.6 Java Source Files**

- **Point 23**

The javadoc is not complete, and is missing for the following methods:

- `getRequirementsForSupplier()`
- `createAutoRequirementsForOrder()`
- `createATPRequirementsForOrder()`
- `updateRequirementsToOrdered()`

### **2.1.7 Output Format**

- **Point 42**

The following lines contain log error messages that do not provide guidance or hints on how to correct the problem:

- Line 235
- Line 237
- Line 325
- Line 327
- Line 352
- Line 354

## **2.2 Issues of EntityEcaCondition.java**

### **2.2.1 Indention**

- **Point 9**

Tabs are used to indent.

### 2.2.2 Braces

- **Point 11**

For the following lines, the single statement to execute is not surrounded by curly braces.

- Line 65

### 2.2.3 File Organization

- **Point 13**

Many lines of code are not broken up properly and exceed 80 characters.

- **Point 14**

For the following lines, the line length exceed 120 characters.

- Line 76
- Line 80

### 2.2.4 Wrapping Lines

- **Point 16**

Higher-level breaks are not used.

### 2.2.5 Java Source Files

- **Point 23**

The javadoc is not complete, and is missing for the following methods:

- `getLValue()`
- `getRValue()`
- `getOperator()`
- `getFieldNames()`

### 3 APPENDIX

#### 3.1 Software and Tools Used

The tools used to creat this document are:

- Github: for version control
- Latex: for typesetting

#### 3.2 Hours of Work

|                   |         |
|-------------------|---------|
| Gao Xiao          | 6 Hours |
| Kang Shuwen       | 6 Hours |
| Liubov Bolshakova | 4 Hours |

## 4 REFERENCES

- AA 2016/2017 Software Engineering 2 - Project goal, schedule and rules
- AA 2016/2017 Software Engineering 2 - Code Inspection Assignment - Task Description
- OFBiz - Apache OFBiz Documentation
- Apache OFBiz Project Open Wiki article
- Wikipedia article on OFBiz [https://en.wikipedia.org/wiki/Apache\\_OFBiz](https://en.wikipedia.org/wiki/Apache_OFBiz)