



## Software Documentation : Java Custom Review Feedback Management v2.0

---

This page last changed on May 15, 2013 by volodymyrk.

### 1. Scope

#### 1.1 Overview

The Review Feedback Management component will define a new entity called "Review Feedback" and provide basic CRUD operations for it.

In version 2.0 the review feedback entity will be refined and separated in two entity classes. The component will also persist audit data for all modifications to the review feedback records.

#### 1.2 Logic Requirements

##### 1.2.1 Review Feedback entities

The component will provide entity classes for the review feedback data. There will be two entity classes:

1. ReviewFeedback. This class defines the general data about the review feedback: project ID, general comment, creation and modification authors and timestamps. This class will also contain a (possibly empty) list of associated ReviewFeedbackDetail instances.
2. ReviewFeedbackDetail. This class defines review feedback for a specific reviewer: reviewer user ID, score and textual feedback.

These records will be stored in the `tcs_catalog.review_feedback` and `tcs_catalog.review_feedback_detail` tables. The DDL for the tables is provided within the DB schema. The interface of all CRUD operations of the component will work with the ReviewFeedback class. The data for ReviewFeedbackDetail will be accessed through ReviewFeedback entity.

Please note that in the DB there's a FK from `review_feedback_detail` table to `review_feedback` table, but in the data model the association is the opposite: the ReviewFeedback will contain a list of associated ReviewFeedbackDetail records.

##### 1.2.2 Create operation

The component will provide a method to create a new Review Feedback entry (along with its details records). The method will perform validation of the field values. The method will also have a string "operator" parameter which will be used for the audit records.

The new entry ID will be generated by the DB sequence. The method will return the newly created entry with all fields set.

##### 1.2.3 Update operation

The component will provide a method to update an existing Review Feedback entry. The method will perform validation of the field values. The method will also have a string "operator" parameter which will be used for the audit records.

The method will return the updated ReviewFeedback instance.

##### 1.2.4 Retrieve operation

The component will provide a method to retrieve an existing Review Feedback record by its ID. If the record with the given ID doesn't exist the method will return null.

The component will also provide a method to retrieve all Review Feedback entries for a given project ID.



### 1.2.5 Delete operation

The component will provide a method to delete an existing Review Feedback entry by its ID. The method's return value will indicate if the entry with the given ID was deleted.

### 1.2.6 Data Audit

The create and update operations will persist audit data in the `tcs_catalog.review_feedback_audit` and `tcs_catalog.review_feedback_detail_audit` tables. For the create operation the audit action type will be "create" for both entities. For the update operation, the audit action type will be "update" for ReviewFeedback entity and one of the following for the ReviewFeedbackDetail entity:

1. "Create" when a review feedback detail is being added for a new reviewer user ID.
2. "Update" when an existing review feedback detail is being updated.
3. "Delete" when an existing review feedback detail is being deleted.

When a review feedback record is being deleted, all associated audit records will be deleted too.

Also, the following fields of the ReviewFeedback entity are considered a part of the audit data: `createUser`, `createDate`, `modifyUser`, `modifyDate`. These fields are not supposed to be set directly by the application code (you might as well remove the setters for them). Instead, they will be set in the create and update methods. The create method will set all four fields and the update method will set only the `modifyUser`, `modifyDate` fields.

### 1.2.7 Thread-Safety

The component is required to be thread-safe.

## 1.3 Required Algorithms

None.

## 1.4 Example of the Software Usage

The component will be used by the Online Review application to manage the review feedback records.

## 1.6 Future Component Direction

**Any enhancement needs to be approved** either in forum or in email with managers to eliminate over-complicating the component with useless functions. All performance optimizations are highly encouraged and do not require explicit approval.

# 2. Interface Requirements

### 2.1.1 Graphical User Interface Requirements

None, only API interface will be provided.

### 2.1.2 External Interfaces

None.

### 2.1.3 Environment Requirements

- Development language: Java 1.5
- Compile target: Java 1.5, Java 1.6



#### **2.1.4 Package Structure**

com.topcoder.management.reviewfeedback

### **3. Software Requirements**

#### **3.1 Administration Requirements**

##### **3.1.1 What elements of the application need to be configurable?**

None.

#### **3.2 Technical Constraints**

##### **3.2.1 Are there particular frameworks or standards that are required?**

None.

##### **3.2.2 TopCoder Software Component Dependencies:**

- Base Exception 2.0
- Configuration API 1.1.0
- Configuration Persistence 1.0.2
- DB Connection Factory 1.1.0
- Logging Wrapper [2.0.0](#)

**\*\*Please review the TopCoder Software component catalog for existing components that can be used in the design.**

##### **3.2.3 Third Party Component, Library, or Product Dependencies:**

Any third party library needs to be approved.

##### **3.2.4 QA Environment:**

- Java 1.5
- RedHat Linux 4
- Windows 2000
- Windows 2003

#### **3.3 Design Constraints**

The component design and development solutions must adhere to the guidelines as outlined in the TopCoder Software Component Guidelines. Modifications to these guidelines for this component should be detailed below.

#### **3.4 Required Documentation**

##### **3.4.1 Design Documentation**

- Use-Case Diagram
- Class Diagram
- Sequence Diagram
- Component Specification



### **3.4.2 Help / User Documentation**

- Design documents must clearly define intended component usage in the 'Documentation' tab of TC UML Tool.