



## **Deliverable Management Requirements Specification**

### **1. Scope**

#### **1.1 Overview**

Various deliverables need to be fulfilled for a project during a specific phase. Usually a phase can be concluded only when all the required deliverables are present. The component defines an API to track the deliverables. The actual mechanism to verify each deliverable will be pluggable.

It also supports two types of specific deliverables, document upload and submission. Submission is one type of upload that will have extra properties.

#### **1.2 Logic Requirements**

##### **1.2.1 Upload Entity**

An upload will be associated with a type, a status, a project, a resource and the parameter to retrieve the uploaded file. A submission is a particular type of upload that will be associated with an extra submission status. Please refer to 2.1.2 for details.

##### **1.2.2 Deliverable Entity**

A deliverable will be associated with a name, a project, a phase, a resource and an optional submission. It will also have a required flag and a completion date. Please refer to 2.1.2 for details.

##### **1.2.3 Upload Management Operations**

Despite manipulation of the upload structure defined above, this component will support the following operations.

###### **1.2.3.1 Create Upload**

A new upload can be created. The identifier will be provided with ID Generator. Upload should be validated according to the defined rules.

The operator needs to be provided for auditing purpose.

###### **1.2.3.2 Update Upload**

An existing upload can be updated. Upload should be validated according to the defined rules.

The operator needs to be provided for auditing purpose.

###### **1.2.3.3 Get Upload**

An upload can be retrieved by specified identifier.

###### **1.2.3.4 Search Uploads**

An array of uploads can be retrieved by specified search criteria.

###### **1.2.3.4.1 Upload Search Criteria**

At minimum, the criteria should be capable of specifying any combination of upload type, status, project and resource.

###### **1.2.3.5 All Upload Types**

There should be a way to retrieve all upload types in the system.

###### **1.2.3.6 All Upload Statuses**

There should be a way to retrieve all upload statuses in the system.



#### 1.2.4 *Submission Management Operations*

Despite manipulation of the submission structure defined above, this component will support the following operations.

##### 1.2.4.1 *Create Submission*

A new submission can be created. The identifier will be provided with ID Generator. Submission should be validated according to the defined rules. The upload associated with the submission is assumed to be already created.

The operator needs to be provided for auditing purpose.

##### 1.2.4.2 *Update Submission*

An existing submission can be updated. Submission should be validated according to the defined rules. The upload associated with the submission will not be updated.

The operator needs to be provided for auditing purpose.

##### 1.2.4.3 *Get Submission*

A submission can be retrieved by specified identifier.

##### 1.2.4.4 *Search Submissions*

An array of submissions can be retrieved by specified search criteria.

##### 1.2.4.4.1 *Submission Search Criteria*

At minimum, the criteria should be capable of specifying any combination of project, resource and submission status.

##### 1.2.4.5 *All Submission Statuses*

There should be a way to retrieve all submission statuses in the system.

#### 1.2.5 *Deliverable Search*

An array of deliverables can be retrieved by specified search criteria

##### 1.2.5.1 *Deliverable Search Criteria*

At minimum, the criteria should be capable of specifying any combination of name, project, phase, submission, resource, required flag and completion flag.

##### 1.2.5.2 *Deliverables Per Submission*

If a deliverable is configured to be per submission, instance of Deliverable should be present for each submission with Active status.

##### 1.2.5.3 *Check Deliverables*

There should be a pluggable mechanism to check if each deliverable is complete and fill the completion date. No concrete implementation is required for this component.

#### 1.2.6 *Search Builder Usage*

Search Builder should be used with the searching functionality. Only the identifiers of the entities should return from the Search Builder. Convenient methods should be provided to create the applicable filters.

#### 1.2.7 *Persistence*

Persistence needs to be pluggable. For this release an Informix plug-in will be developed. The SQL scripts will be provided.

##### 1.2.7.1 *Persistence Implementation*

The persistence implementation needs to be designed in this component, but will be separated



into a second development project. Please put all persistence implementation related information into a separate sub-package and clearly mark the responsibilities of the two development projects.

### 1.3 Required Algorithms

No specific algorithms are required.

### 1.4 Example of the Software Usage

A project management application can use the component to track all the uploaded deliverables and automate project execution. Once all required deliverables are fulfilled the project can be advanced to the next phase.

### 1.5 Future Component Direction

None.

## 2. Interface Requirements

### 2.1.1 Graphical User Interface Requirements

None.

### 2.1.2 External Interfaces

Design must adhere to the interface diagram definition. Designer can choose to add more methods to the classes/interfaces, but must keep the ones defined on the diagram as a minimum. Source files can be found in the distribution.

#### 2.1.2.1 Upload Type

An upload will be associated with a type. An upload type consists of

ID	a numeric identifier
Name	name of the upload type for display purpose

#### 2.1.2.2 Upload Status

An upload will be associated with a status. An upload status consists of

ID	a numeric identifier
Name	name of the upload type for display purpose

#### 2.1.2.3 Upload

An upload consists of

ID	a numeric identifier
Upload Type	each upload will have a type
Upload Status	each upload will have a status
Project	the project the upload is associated with, could be represented by a numeric identifier
Resource	the person that owns the upload, could be represented by a numeric Identifier
Parameter	the parameter that identifies the uploaded file, should be a String

#### 2.1.2.4 Submission Status

A submission will be associated with a status. A submission status consists of

ID	a numeric identifier
Name	name of the submission status for display purpose

#### 2.1.2.5 Submission

A submission consists of

ID	a numeric identifier
Upload	the latest upload for the submission
Status	each submission will have a status



#### 2.1.2.6 Deliverable

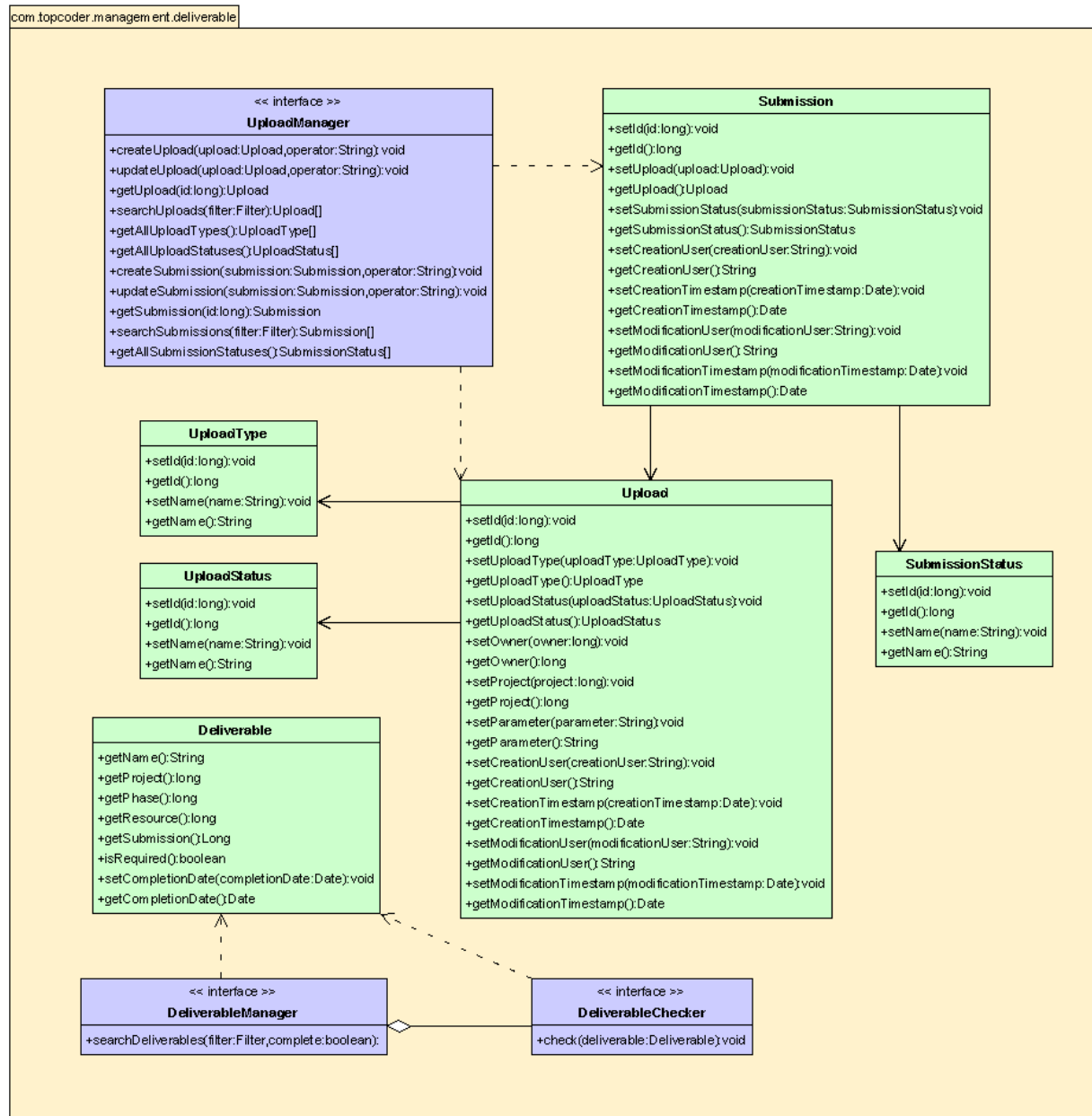
A deliverable is a piece of work a resource needs to fulfill in a certain phase of the project. It consists of

Name	name of the deliverable for display purpose
Project	a numeric identifier to the project the deliverable is associated with
Phase	a numeric identifier to the phase the deliverable is associated with
Submission	a numeric identifier to the submission the deliverable is associated with
Resource	a numeric identifier to the resource that produces the deliverable
Required	a Boolean flag indicating whether the deliverable is required for the phase to end
Completion Date	the date on which the deliverable is completed. If a completion date does not exist, then the deliverable is still pending.

#### 2.1.2.7 Auditing Fields

Upload and Submission must also include auditing fields of creation/modification operator and timestamp. These fields will not be provided by component users.

## Deliverable Management Interface Diagram



Created with Poseidon for UML Community Edition. Not for Commercial Use.

### 2.1.3 Environment Requirements

- Development language: Java1.4
- Compile target: Java1.4

### 2.1.4 Package Structure

com.topcoder.management.deliverable

### 3. Software Requirements

#### 3.1 Administration Requirements

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

- Database connection
- Deliverable check mechanisms

#### 3.2 Technical Constraints

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

JDBC

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

- Configuration Manager
- DB Connection Factory
- ID Generator
- Search Builder

**\*\*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:*

None.

##### 3.2.4 *QA Environment:*

- Solaris 7
- RedHat Linux 7.1
- Windows 2000
- Windows 2003
- Informix 10.0

#### 3.3 Design Constraints

The component design and development solutions must adhere to the guidelines as outlined in the TopCoder Software Component Guidelines.

##### 3.3.1 *Database Connections*

Database connections must not be cached within the component. Connections should be created for each operation and closed afterwards.

##### 3.3.2 *Component Scalability*

The component needs to be scalable. Running multiple instances in the same JVM or in multiple JVM's concurrently should not cause any problem.

#### 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 Poseidon.