[TopCoder]

Late Deliverables Management 1.0 Component Specification

1. Design

TopCoder has an utility called Late Deliverables Tracker that periodically examines all active projects in the Online Review and records when the deliverables are late (e.g. review scorecards, final fixes etc.). Late Deliverables Management provides search and update functionality for the late deliverables. Different search filters are supported such as filtering by the project status, project category, deliverable type etc. It is also possible to combine the filters with AND and OR logical operators.

This component provides LateDeliverableManager interface together with its implementation that uses pluggable LateDeliverablePersistence instance for updating late deliverables and Search Builder component for retrieving or searching for late deliverables.

Also this component provides a static helper class LateDeliverableFilterBuilder that defines method for creating filters that can be used when searching for late deliverables.

1.1 Design Patterns

Strategy pattern – LateDeliverableManager and its implementation can be used in some external strategy context; LateDeliverableManagerImpl uses pluggable LateDeliverablePersistence instance.

Delegate pattern – LateDeliverableManagerImpl#update() method simply delegates execution to the namesake method of the pluggable LateDeliverablePersistence implementation instance.

DAO/DTO pattern – LateDeliverableManager and LateDeliverablePersistence are DAOs for LateDeliverable DTO.

1.2 Industry Standards

SQL, JDBC, JavaBeans

1.3 Required Algorithms

1.3.1 Logging

This component must perform logging in all public business methods of LateDeliverableManagerImpl and DatabaseLateDeliverablePersistence.

All information described below must be logged using log:Log attribute. If log attribute is null, then logging is not required to be performed.

In all mentioned methods method entrance with input argument, method exit with return value and call duration time must be logged at DEBUG level. It's not required to log method exit when method throws an exception.

All thrown exceptions and errors must be logged at ERROR level.

1.4 Component Class Overview

DatabaseLateDeliverablePersistence

This class is an implementation of LateDeliverablePersistence that updates late deliverables in database persistence using JDBC and DB Connection Factory component. This class uses Logging Wrapper component to log errors and debug information.

LateDeliverable

This class is a container for information about a single late deliverable. It is a simple JavaBean (POJO) that provides getters and setters for all private attributes and performs no argument validation in the setters.

LateDeliverableFilterBuilder

This is a static helper class that provides factory methods for creating filters that can be used when searching for late deliverables using LateDeliverableManagerImpl and possibly other implementations of LateDeliverableManager.

LateDeliverableManager [interface]

This interface represents a late deliverable manager. It defines methods for updating/retrieving late deliverable and searching for late deliverables that are matched with the given filter and optionally are restricted to the specified user.

LateDeliverableManagerImpl

This class is an implementation of LateDeliverableManager that uses Search Builder component to retrieve by ID or search for late deliverables in persistence and pluggable LateDeliverablePersistence instance to update late deliverables in persistence. This class uses Logging Wrapper component to log errors and debug information.

LateDeliverablePersistence [interface]

This interface represents a late deliverable persistence. Currently it defines just a single method for updating late deliverable in persistence. Other methods can be added in future.

1.5 Component Exception Definitions

LateDeliverableManagementConfigurationException

This exception is thrown by LateDeliverableManagerImpl and implementations of LateDeliverablePersistence when some error occurs while initializing an instance using the given configuration.

LateDeliverableManagementException

This exception is thrown by implementations of LateDeliverableManager when some not expected error occurred. Also this exception is used as a base class for other specific custom exceptions.

LateDeliverableNotFoundException

This exception is thrown by LateDeliverableManagerImpl and implementations of LateDeliverablePersistence when late deliverable with the specified ID doesn't exist in persistence.

LateDeliverablePersistenceException

This exception is thrown by LateDeliverableManagerImpl and implementations of LateDeliverablePersistence when some error occurs while accessing the persistence. Also this exception is used as a base class for other specific custom exceptions.

1.6 Thread Safety

This component is thread safe.

Implementations of LateDeliverableManager and LateDeliverablePersistence are required to be thread safe when entities passed to them are used by the caller in thread safe manner. Additionally it's assumed that configure() method of LateDeliverablePersistence implementations will be called just once right after instantiation.

LateDeliverableManagerImpl is immutable and thread safe when entities passed to it are used by the caller in thread safe manner. It uses thread safe SearchBundle, LateDeliverablePersistence and Log instances.

DatabaseLateDeliverablePersistence is mutable, but thread safe when configure() method is called just once right after instantiation and entities passed to it are used by the caller in thread safe manner. It uses thread safe DBConnectionFactory and Log instances.
DatabaseLateDeliverablePersistence uses transactions when updating data in the database.

LateDeliverable is mutable and not thread safe entity.

LateDeliverableFilterBuilder is immutable and thread safe static utility class.

2. Environment Requirements

2.1 Environment

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

QA Environment: Java 1.5, RedHat Linux 4, Windows 2000, Windows 2003

2.2 TopCoder Software Components

Base Exception 2.0 – is used by custom exceptions defined in this component.

Configuration API 1.0 – is used for initializing classes from this component.

Configuration Persistence 1.0.2 – is used for reading configuration from file.

Search Builder 1.3.1 – is used for searching for late deliverables in persistence.

Database Abstraction 1.1 – defines CustomResultSet class used in this component.

DB Connection Factory 1.1 – is used for creating database connections.

Logging Wrapper 1.2 – is used for logging errors and debug information.

Object Factory 2.0.1 – is used for creating pluggable object instances.

Object Factory Configuration API Plugin 1.0 – allows to use Configuration API for creating Object Factory.

NOTE: The default location for TopCoder Software component jars is../lib/tcs/COMPONENT_NAME/COMPONENT_VERSION relative to the component installation. Setting the tcs_libdir property in topcoder_global.properties will overwrite this default location.

2.3 Third Party Components

None

3. Installation and Configuration

3.1 Package Name

com.topcoder.management.deliverable.late
com.topcoder.management.deliverable.late.impl
com.topcoder.management.deliverable.late.impl.persistence
com.topcoder.management.deliverable.late.search

3.2 Configuration Parameters

3.2.1 Configuration of LateDeliverableManagerImpl

The following table describes the structure of ConfigurationObject passed to the constructor of LateDeliverableManagerImpl class (angle brackets are used for identifying child configuration objects). This ConfigurationObject can be optionally read from a configuration file using Configuration Persistence component.

Parameter	Description	Values
loggerName	The name of Logging Wrapper logger to be used for logging errors and debug information. When not provided, logging is not performed.	String. Not empty. Optional.
searchBundleManager Namespace	The namespace used when creating an instance of SearchBundleManager.	String. Not empty. Required.
nonRestrictedSearchBundle Name	The name of the search bundle used by this class when searching for late deliverables with no user restriction.	String. Not empty. Required.

restrictedSearchBundleName	The name of the search bundle used by this class when searching for late deliverables with specific user restriction.	String. Not empty. Required.
<objectFactoryConfig>	This section contains configuration of Object Factory used by this class for creating pluggable object instances.	ConfigurationObject. Required.
persistenceKey	The Object Factory key that is used for creating an instance of LateDeliverablePersistence to be used by this manager.	String. Not empty. Required.
<persistenceConfig>	The configuration for LateDeliverablePersistence instance.	ConfigurationObject. Required.

3.2.2 *Configuration of DatabaseLateDeliverablePersistence*

The following table describes the structure of ConfigurationObject passed to the constructor of DatabaseLateDeliverablePersistence class (angle brackets are used for identifying child configuration objects).

Parameter	Description	Values
loggerName	The name of Logging Wrapper logger to be used for logging errors and debug information. When not provided, logging is not performed.	String. Not empty. Optional.
dbConnectionFactoryConfig	The configuration to be used for creating DBConnectionFactoryImpl instance.	ConfigurationObject. Required.
connectionName	The connection name to be passed to the connection factory when establishing a database connection. If not specified, a default connection is used.	String. Not empty. Optional.

3.3 Dependencies Configuration

Please see docs of Logging Wrapper, DB Connection Factory and Object Factory components to configure them properly.

3.3.1 *Configuration of Search Builder component*

The proposed configuration for Search Builder component is provided below:

```
<?xml version="1.0"?>
<CMConfig>
  <Config name="LateDeliverableManagerImpl.SearchBuilderManager">
    <Property name="searchStrategyFactoryNamespace">
      <Value>com.topcoder.search.builder.strategy.factory</Value>
    </Property>
    <Property name="fieldValidatorFactoryNamespace">
      <Value>com.topcoder.search.builder.validator.factory</Value>
    </Property>
    <Property name="searchBundles">
      <Property name="Non-restricted Late Deliverable Search Bundle">
        <Property name="searchStrategy">
          <Property name="class">
            <Value>dbStrategy</Value>
          </Property>
        </Property>
        <Property name="context">
          <Value>
```

```
    SELECT lt.late_deliverable_id,
      lt.project_phase_id,
      lt.resource_id,
      lt.deliverable_id,
      lt.deadline,
      lt.create_date,
      lt.forgive_ind,
      lt.last_notified,
      lt.delay,
      lt.explanation,
      lt.response
    FROM late_deliverable lt
      INNER JOIN project_phase pp ON pp.project_phase_id = lt.project_phase_id
      INNER JOIN project p ON p.project_id = pp.project_id
    WHERE
  </Value>
</Property>
<Property name="searchableFields">
  <Property name="dummy">
    <Property name="validator">
      <Property name="class">
        <Value>validator</Value>
      </Property>
      <Property name="identifier">
        <Value>null</Value>
      </Property>
    </Property>
  </Property>
</Property>
<Property name="alias">
  <Property name="id">
    <Value>lt.late_deliverable_id</Value>
  </Property>
  <Property name="projectPhaseId">
    <Value>lt.project_phase_id</Value>
  </Property>
  <Property name="resourceId">
    <Value>lt.resource_id</Value>
  </Property>
  <Property name="deliverableId">
    <Value>lt.deliverable_id</Value>
  </Property>
  <Property name="deadline">
    <Value>lt.deadline</Value>
  </Property>
  <Property name="createDate">
    <Value>lt.create_date</Value>
  </Property>
  <Property name="forgiven">
    <Value>lt.forgive_ind</Value>
  </Property>
  <Property name="lastNotified">
    <Value>lt.last_notified</Value>
  </Property>
  <Property name="delay">
    <Value>lt.delay</Value>
  </Property>
  <Property name="explanation">
    <Value>lt.explanation</Value>
  </Property>
  <Property name="response">
    <Value>lt.response</Value>
  </Property>
  <Property name="projectId">
    <Value>pp.project_id</Value>
  </Property>
  <Property name="projectStatusId">
    <Value>p.project_status_id</Value>
  </Property>
  <Property name="projectCategoryId">
    <Value>p.project_category_id</Value>
```

```
    </Property>
  </Property>
</Property>

<Property name="Restricted Late Deliverable Search Bundle">
  <Property name="searchStrategy">
    <Property name="class">
      <Value>dbStrategy</Value>
    </Property>
  </Property>
  <Property name="context">
    <Value>
      SELECT DISTINCT lt.late_deliverable_id,
        lt.project_phase_id,
        lt.resource_id,
        lt.deliverable_id,
        lt.deadline,
        lt.create_date,
        lt.forgive_ind,
        lt.last_notified,
        lt.delay,
        lt.explanation,
        lt.response
      FROM late_deliverable lt
        INNER JOIN project_phase pp
          ON pp.project_phase_id = lt.project_phase_id
        INNER JOIN project p ON p.project_id = pp.project_id
        LEFT OUTER JOIN resource r
          ON r.project_id = p.project_id and r.resource_role_id IN (13, 14, 15)
        LEFT OUTER JOIN resource_info ri
          ON ri.resource_id = r.resource_id and ri.resource_info_type_id = 1
        INNER JOIN resource_info ri2
          ON ri2.resource_id = lt.resource_id and ri2.resource_info_type_id = 1
        LEFT OUTER JOIN corporate_oltp:tc_direct_project d
          ON d.project_id = p.tc_direct_project_id
        LEFT OUTER JOIN corporate_oltp:user_permission_grant g
          ON g.resource_id = d.project_id
      WHERE
    </Value>
  </Property>
  <Property name="searchableFields">
    <Property name="dummy">
      <Property name="validator">
        <Property name="class">
          <Value>validator</Value>
        </Property>
        <Property name="identifier">
          <Value>null</Value>
        </Property>
      </Property>
    </Property>
  </Property>
  <Property name="alias">
    <Property name="id">
      <Value>lt.late_deliverable_id</Value>
    </Property>
    <Property name="projectPhaseId">
      <Value>lt.project_phase_id</Value>
    </Property>
    <Property name="resourceId">
      <Value>lt.resource_id</Value>
    </Property>
    <Property name="deliverableId">
      <Value>lt.deliverable_id</Value>
    </Property>
    <Property name="deadline">
      <Value>lt.deadline</Value>
    </Property>
    <Property name="createDate">
      <Value>lt.create_date</Value>
    </Property>
```

```
            <Property name="forgiven">
              <Value>lt.forgive_ind</Value>
            </Property>
            <Property name="lastNotified">
              <Value>lt.last_notified</Value>
            </Property>
            <Property name="delay">
              <Value>lt.delay</Value>
            </Property>
            <Property name="explanation">
              <Value>lt.explanation</Value>
            </Property>
            <Property name="response">
              <Value>lt.response</Value>
            </Property>
            <Property name="projectId">
              <Value>pp.project_id</Value>
            </Property>
            <Property name="projectStatusId">
              <Value>p.project_status_id</Value>
            </Property>
            <Property name="projectCategoryId">
              <Value>p.project_category_id</Value>
            </Property>
            <Property name="managerUserId">
              <Value>ri.value</Value>
            </Property>
            <Property name="lateUserId">
              <Value>ri2.value</Value>
            </Property>
            <Property name="tcDirectUserId">
              <Value>g.user_id</Value>
            </Property>
          </Property>
        </Property>

      </Property>
    </Config>
    <Config name="com.topcoder.search.builder.validator.factory">
      <Property name="validator:null">
        <Property name="type">
          <Value>com.topcoder.util.datavalidator.NullValidator</Value>
        </Property>
      </Property>
    </Config>
    <Config name="com.topcoder.search.builder.strategy.factory">
      <Property name="dbStrategy">
        <Property name="type">
          <Value>com.topcoder.search.builder.database.DatabaseSearchStrategy</Value>
        </Property>
        <Property name="params">
          <Property name="param1">
            <Property name="type">
              <Value>String</Value>
            </Property>
            <Property name="value">
              <Value>DBSearchStrategy</Value>
            </Property>
          </Property>
        </Property>
      </Property>
    </Config>
    <Config name="DBSearchStrategy">
      <!-- Property defining a specification for constructing the dbConnectionFactory to use. -
->
      <Property name="connectionFactory">

        <!-- The namespace of the ConnectionFactory -->
        <Property name="name">
          <Value>com.topcoder.db.connectionfactory.DBConnectionFactoryImpl</Value>
        </Property>
```

```
      <Property name="class">
        <Value>com.topcoder.db.connectionfactory.DBConnectionFactoryImpl</Value>
      </Property>
    </Property>


    <!--
                    The name to request from the connection factory when acquiring a
connection. If not present, then the default connection is used.
                     -->
    <Property name="connectionName">
      <Value>dbconnection</Value>
    </Property>


    <Property name="searchFragmentFactoryNamespace">
      <Value>com.topcoder.search.builder.database.factory</Value>
    </Property>

    <Property name="searchFragmentBuilders">
      <Property name="first">
        <Property name="targetFilter">
          <Value>com.topcoder.search.builder.filter.AndFilter</Value>
        </Property>
        <Property name="className">
          <Value>com.topcoder.search.builder.database.AndFragmentBuilder</Value>
        </Property>
      </Property>

      <Property name="second">
        <Property name="targetFilter">
          <Value>com.topcoder.search.builder.filter.OrFilter</Value>
        </Property>
        <Property name="className">
          <Value>com.topcoder.search.builder.database.OrFragmentBuilder</Value>
        </Property>
      </Property>

      <Property name="third">
        <Property name="targetFilter">
          <Value>com.topcoder.search.builder.filter.LikeFilter</Value>
        </Property>
        <Property name="className">
          <Value>com.topcoder.search.builder.database.LikeFragmentBuilder</Value>
        </Property>
      </Property>

      <Property name="fourth">
        <Property name="targetFilter">
          <Value>com.topcoder.search.builder.filter.NotFilter</Value>
        </Property>
        <Property name="className">
          <Value>com.topcoder.search.builder.database.NotFragmentBuilder</Value>
        </Property>
      </Property>

      <Property name="fifth">
        <Property name="targetFilter">
          <Value>com.topcoder.search.builder.filter.EqualToFilter</Value>
        </Property>
        <Property name="className">
          <Value>com.topcoder.search.builder.database.EqualsFragmentBuilder</Value>
        </Property>
      </Property>

      <Property name="sixth">
        <Property name="targetFilter">
          <Value>com.topcoder.search.builder.filter.InFilter</Value>
        </Property>
        <Property name="className">
          <Value>com.topcoder.search.builder.database.InFragmentBuilder</Value>
```

```
        </Property>
      </Property>

      <Property name="eighth">
        <Property name="targetFilter">
          <Value>com.topcoder.search.builder.filter.NullFilter</Value>
        </Property>
        <Property name="className">
          <Value>com.topcoder.search.builder.database.NullFragmentBuilder</Value>
        </Property>
      </Property>

      <Property name="ninth">
        <Property name="targetFilter">
          <Value>com.topcoder.search.builder.filter.GreaterThanFilter</Value>
        </Property>
        <Property name="className">
          <Value>com.topcoder.search.builder.database.RangeFragmentBuilder</Value>
        </Property>
      </Property>

      <Property name="tenth">
        <Property name="targetFilter">
          <Value>com.topcoder.search.builder.filter.GreaterThanOrEqualToFilter</Value>
        </Property>
        <Property name="className">
          <Value>com.topcoder.search.builder.database.RangeFragmentBuilder</Value>
        </Property>
      </Property>

      <Property name="eleventh">
        <Property name="targetFilter">
          <Value>com.topcoder.search.builder.filter.BetweenFilter</Value>
        </Property>
        <Property name="className">
          <Value>com.topcoder.search.builder.database.RangeFragmentBuilder</Value>
        </Property>
      </Property>

      <Property name="twelvth">
        <Property name="targetFilter">
          <Value>com.topcoder.search.builder.filter.LessThanOrEqualToFilter</Value>
        </Property>
        <Property name="className">
          <Value>com.topcoder.search.builder.database.RangeFragmentBuilder</Value>
        </Property>
      </Property>

      <Property name="thirteenth">
        <Property name="targetFilter">
          <Value>com.topcoder.search.builder.filter.LessThanFilter</Value>
        </Property>
        <Property name="className">
          <Value>com.topcoder.search.builder.database.RangeFragmentBuilder</Value>
        </Property>
      </Property>
    </Property>
  </Config>
  <Config name="com.topcoder.search.builder.database.factory">
  </Config>
</CMConfig>
```

4. Usage Notes

4.1 Required steps to test the component

- Extract the component distribution.
- Follow Dependencies Configuration.

- Execute 'ant test' within the directory that the distribution was extracted to.

4.2 Required steps to use the component

Please see the demo.

4.3 Demo

4.3.1 Sample LateDeliverableManagerImpl configuration file

```
<?xml version="1.0"?>
<CMConfig>
  <Config name="com.topcoder.management.deliverable.late.impl.LateDeliverableManagerImpl">
    <Property name="loggerName">
      <Value>myLogger</Value>
    </Property>
    <Property name="objectFactoryConfig">
      <!-- Put Object Factory configuration here -->
    </Property>
    <Property name="searchBundleManagerNamespace">
      <Value>LateDeliverableManagerImpl.SearchBuilderManager</Value>
    </Property>
    <Property name="nonRestrictedSearchBundleName">
      <Value>Non-restricted Late Deliverable Search Bundle</Value>
    </Property>
    <Property name="restrictedSearchBundleName">
      <Value>Restricted Late Deliverable Search Bundle</Value>
    </Property>
    <Property name="persistenceKey">
      <Value>DatabaseLateDeliverablePersistence</Value>
    </Property>
    <Property name="persistenceConfig">
      <Property name="loggerName">
        <Value>myLogger</Value>
      </Property>
      <Property name="dbConnectionFactoryConfig">
        <!-- Put DB Connection Factory configuration here -->
      </Property>
      <Property name="connectionName">
        <Value>myConnection</Value>
      </Property>
    </Property>
  </Config>
</CMConfig>
```

4.3.2 Sample input and output

Assume that initially database contains the following data:

late_deliverable table
(not important columns are skipped)

late_deliverable_id	project_phase_id	resource_id	deliverable_id	deadline	forgive_ind	explanation
1	101	1001	4 (Review Scorecard)	2010-11-22 09:05:00	0	NULL
2	102	1002	3 (Screening Scorecard)	2010-11-25 12:00:00	0	NULL

project_phase table
(not important columns are skipped)

project_phase_id	project_id
101	100000
102	100001

project table
(not important columns are skipped)

project_id	project_status_id	project_category_id

100000	7 (Completed)	1 (Design)
100001	1 (Active)	1 (Design)

resource table
(not important columns are skipped)

resource_id	resource_role_id	project_id	project_phase_id
1001	4 (Reviewer)	100000	101
1002	3 (Screener)	100001	102
1003	13 (Manager)	100001	NULL

resource_info table
(not important columns are skipped)

resource_id	resource_info_type_id	value
1001	1	1
1002	1	2
1003	1	3

Then the following code can be executed:

```
// Create an instance of LateDeliverableManagerImpl using custom configuration
ConfigurationObject configuration = ...
LateDeliverableManagerImpl lateDeliverableManager = new
LateDeliverableManagerImpl(configuration);

// Create an instance of LateDeliverableManagerImpl using custom config file and namespace
lateDeliverableManager = new LateDeliverableManagerImpl("config.properties", "my_namespace");

// Create an instance of LateDeliverableManagerImpl using default config file
lateDeliverableManager = new LateDeliverableManagerImpl();

// Retrieve the late deliverable with ID=1
LateDeliverable lateDeliverable = lateDeliverableManager.retrieve(1);
// lateDeliverable.getId() must be 1
// lateDeliverable.getProjectPhaseId() must be 101
// lateDeliverable.getResourceId() must be 1001
// lateDeliverable.getDeliverableId() must be 4
// lateDeliverable.isForgiven() must be false
// lateDeliverable.getExplanation() must be null

// Update the late deliverable by changing its forgiven flag and explanation
lateDeliverable.setForgiven(true);
lateDeliverable.setExplanation("OR didn't work");
lateDeliverableManager.update(lateDeliverable);
```

After this step the database must be updated respectively:

late_deliverable table
(not important columns are skipped)

late_deliverable_id	project_phase_id	resource_id	deliverable_id	deadline	forgive_ind	explanation
1	101	1001	4 (Review Scorecard)	2010-11-22 09:05:00	1	OR didn't work
2	102	1002	3 (Screening Scorecard)	2010-11-25 12:00:00	0	NULL

```
// Search for all forgiven late deliverables for project with ID=100000
Filter forgivenFilter = LateDeliverableFilterBuilder.createForgivenFilter(true);
Filter projectIdFilter = LateDeliverableFilterBuilder.createProjectIdFilter(100000);
Filter compositeFilter = new AndFilter(forgivenFilter, projectIdFilter);
List<LateDeliverable> lateDeliverables =
    lateDeliverableManager.searchAllLateDeliverables(compositeFilter);
// lateDeliverables.size() must be 1
// lateDeliverables.get(0).getId() must be 1
// lateDeliverables.get(0).getProjectPhaseId() must be 101
```

```
// lateDeliverables.get(0).getResourceId() must be 1001
// lateDeliverables.get(0).getDeliverableId() must be 4
// lateDeliverables.get(0).isForgiven() must be true
// lateDeliverables.get(0).getExplanation() must be "OR didn't work"

// Search for all late deliverables from design category for all active projects
// to which user with ID=3 has a manager/copilot access
Filter categoryFilter = LateDeliverableFilterBuilder.createProjectCategoryIdFilter(1);
Filter activeProjectFilter = LateDeliverableFilterBuilder.createProjectStatusIdFilter(1);
compositeFilter = new AndFilter(categoryFilter, activeProjectFilter);
lateDeliverables =
    lateDeliverableManager.searchRestrictedLateDeliverables(compositeFilter, 3);
// lateDeliverables.size() must be 1
// lateDeliverables.get(0).getId() must be 2
// lateDeliverables.get(0).getProjectPhaseId() must be 102
// lateDeliverables.get(0).getResourceId() must be 1002
// lateDeliverables.get(0).getDeliverableId() must be 3
// lateDeliverables.get(0).isForgiven() must be false
// lateDeliverables.get(0).getExplanation() must be null
```

5. Future Enhancements

None