Software Documentation : Java Custom Late Deliverables Tracker

This page last changed on Aug 24, 2010 by volodymyrk.

1. Scope

1.1 Overview

Late Deliverables Tracker makes use of the API defined by the Project Management, Phase Management and Deliverable Management components to watch the active projects and make records when the deliverables are late in the Online Review (e.g. review scorecard, final fixes etc.). The projects will be examined periodically.

1.2 Logic Requirements

1.2.1 "Track Late Deliverables" property

Late Deliverables Tracker will only examine the projects that satisfy the following conditions:

1. The project is active (i.e. has "Active" status)
2. There is at least one late phase.
3. The "Track Late Deliverables" property has value "true" in the project extended properties.

1.2.2 Time Interval

Late Deliverables Tracker will examine the projects at a configurable interval. The default value will be 5 minutes. It should run at every interval. Designer should use the TopCoder Job Scheduling component to achieve this purpose.

1.2.3 Saving records

A deliverable is considered late when it is not yet completed and the respective phase is late. When a late deliverable is found a record is added to the tcs_catalog.late_deliverable table.

The following information needs to be saved:

1. Project phase ID
2. Resource ID
3. Deliverable ID
4. Deadline
5. Timestamp when the record was added
6. "Forgive" flag, should be 0 by default.

The list of deliverable IDs to be tracked will be configurable.

The component should not create records if the particular lateness has already been logged. It is also possible that there is more than one record for the same project phase ID and resource ID. This can happen, for example, when the deadline was extended after the resource had been late but then the resource is late again for the extended deadline. To meet the two requirements above the following logic should be applied when a late deliverable is detected:

1. If there is not yet a record for the project phase ID and resource ID add one.
2. If there's already at least one record for the project phase ID and resource ID find the one with the largest deadline value. If this deadline value is greater or equal than the current deadline then don't add the record. Otherwise add a new record with the new deadline.

1.2.4 Warning emails

When a record is added for the late deliverable a warning email should be sent to the corresponding member. The content of the email will be created based on a configurable email template with the following fields:

Copyright TopCoder, Inc. 2008

1. Project Name
2. Project Version
3. Project ID
4. Phase Name
5. Deliverable Name
6. Deadline

The emails should be sent in HTML format.

The email template for each deliverable ID should be configurable (but it should be possible to set a common email template for all deliverable IDs). It should also be possible to turn off email notifications for specific deliverable IDs. Please note that if a deliverable ID is not being tracked (see 1.2.3) the warning email for it won't be sent anyway.

1.2.5 Command Line

The component should provide a command line interface.

1.2.7 Thread-Safety

The component is not required to be thread-safe (if otherwise is not required by the Job Scheduling component).

Special care should be taken though to make sure component works if the time to process all projects is larger than the scheduled period (i.e. when the jobs can overlap in time).

1.2.8 Performance

Special care should be taken to make sure the component is fast:

1. Instead of retrieving all deliverables and then filtering by their IDs it is better to filter by the deliverable IDs in the first place (see DeliverableFilterBuilder#createDeliverableIdFilter method)
2. Instead of retrieving deliverables for all phases and then choosing only those that are for late phases it is better to filter by the late phase IDs in the first place (see DeliverableFilterBuilder#createPhaseIdFilter method)
3. Instead of retrieving all deliverables and then filtering by their status (i.e. completed vs not completed) it is better to retrieve not completed deliverables in the first place (see *complete* parameter of the DeliverableManager's methods).
4. Instead of retrieving late deliverables for each project it is better to retrieve all late deliverables for all projects in a single call. This can be done by pre-collecting the list of the late phase IDs for all projects which are to be processed and then making a **single** call to DeliverableManagement for all projects instead of one call per project.

1.3 Required Algorithms

Designers need to describe the following algorithms:

1. Algorithm to retrieve the list of projects to be inspected.
2. Algorithm to get all late deliverables for a project.
3. Algorithm to decide if the record needs to be added.

1.4 Example of the Software Usage

The component will be used to regularly monitor all active contests in the Online Review and track who is late. The collected information will later be used to apply penalties for the members who are constantly late.

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.

Copyright TopCoder, Inc. 2008

2. Interface Requirements

2.1.1 Graphical User Interface Requirements

None, only API interface and command line 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.deliverable.latetracker

3. Software Requirements

3.1 Administration Requirements

3.1.1 What elements of the application need to be configurable?

- Time interval the jobs are scheduled at.
- Warning email templates for each deliverable type ID. The templates should be read form text files.
- Deliverable IDs to track.
- Deliverable IDs to send warning emails for.

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 1.0
- Configuration Persistence 1.0.2
- Email Engine 3.1.0
- Document Generator 3.1.0
- Project Management 1.0.1
- Project Management Persistence 1.1.2
- Phase Management 1.0.4
- Phase Management Persistence 1.0.2
- Deliverable Management 1.1.1
- Deliverable Management Persistence 1.1.1
- Online Review Deliverables 1.0.2
- Job Scheduling 3.2.0
- Search Builder 1.3.1
- Command Line Utility 1.0.0
- DB Connection Factory 1.1.0

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

Copyright TopCoder, Inc. 2008

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.

Copyright TopCoder, Inc. 2008