[TopCoder]

Late Deliverables Tracker 1.3 Component Specification

1. Design

All changes performed when synchronizing documentation with the version 1.2 of the source code of this component are marked with **purple**.

All changes made in the version 1.3 are marked with **blue**.

All new items in the version 1.3 are marked with **red**.

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 can be examined periodically with use of a command line utility.

Late Deliverables Tracker examines only the projects that satisfy the following conditions:

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

This component performs auditing of all late deliverables in tcs_catalog.late_deliverable table. Additionally it sends warning email messages to the users who have late deliverables. The component can be configured to notify users periodically until the late deliverable is complete.

In the version 1.1 the component was enhanced to track by how much time each deliverable is late. It also accounts for the timeline changes due to the prematurely ended phases.

The version 1.2 enhances the component to regularly email the project managers about not responded late deliverables which are waiting for the resolution. It also fixes the inconsistencies in how the real deadlines vs compensated deadlines are handled. Additionally some new fields are supported in email message templates.

Changes in the version 1.3:

- Added differentiation of late deliverables by type. LateDeliverableType enumeration is added for this.
- In addition to “Missed Deadline” late deliverables supported in the previous version, added support of “Rejected Final Fix” late deliverables. “Rejected Final Fix” late deliverable can be assigned for all Final Fix phases that go directly after Final Review phase (because of rejected previous Final Fix submission).

1.1 Design Patterns

Strategy pattern – LateDeliverablesTracker uses pluggable instances of LateDeliverablesRetriever and LateDeliverableProcessor implementations as strategies. LateDeliverablesRetrieverImpl uses pluggable ProjectManager, PhaseManager and DeliverableManager implementation instances. LateDeliverableProcessorImpl uses pluggable UserRetrieval and ResourceManager implementation instances. NotRespondedLateDeliverablesNotifier uses pluggable LateDeliverableManager, ResourceManager, UserRetrieval, ProjectManager, PhasePersistence and DeliverablePersistence implementation instances. LateDeliverablesTrackingJobRunner and NotRespondedLateDeliverablesNotificationJobRunner are strategy implementations that are used in the strategy context of JobScheduling component.

DAO/DTO pattern – LateDeliverablesRetriever works as a DAO for LateDeliverable DTO, though provides just a read operation.

1.2 Industry Standards

JDBC, SQL, JavaBeans, HTML

1.3 Required Algorithms

1.3.1 Logging

This component must perform logging in:

- main() method of LateDeliverablesTrackingUtility;
- execute() method of LateDeliverablesTracker;
- execute() method of NotRespondedLateDeliverablesNotifier;
- run() and close() methods of LateDeliverablesTrackingJobRunner;
- run() and close() methods of NotRespondedLateDeliverablesNotificationJobRunner;
- retrieve() method of LateDeliverablesRetrieverImpl;
- processLateDeliverable() method of LateDeliverableProcessorImpl;
- sendEmail() method of EmailSendingUtility.

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.

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

Additionally the following information must be logged at INFO level:

- LateDeliverablesTracker: timestamp for each tracking start;
- LateDeliverablesRetrieverImpl: IDs of all active projects and IDs of projects that have late deliverables;
- LateDeliverableProcessorImpl: data for records added to late_deliverables table;
- EmailSendingUtility: timestamp and recipient for each sent email message.
- NotRespondedLateDeliverablesNotifier: the number and IDs of explained but not responded late deliverables.

1.3.2 Retrieval of the current late deliverables

Please see implementation notes of LateDeliverablesRetrieverImpl#retrieve() method in TCUML for a detailed description of the algorithms used for retrieving late deliverables.

1.3.3 Format of email templates used in this component

When sending email messages to the users and managers, this component generates email message title and body from templates using Document Generator component. Please see docs of Document Generator component for additional details about the supported template format.

1.3.3.1 Templates of notifications for users that have late deliverables

These templates correspond to the following fields of LateDeliverableProcessorImpl: missedDeadlineEmailSubjectTemplateTexts, missedDeadlineEmailBodyTemplatePaths, defaultMissedDeadlineEmailSubjectTemplateText, defaultMissedDeadlineEmailBodyTemplatePath, rejectedFinalFixEmailSubjectTemplateText and rejectedFinalFixEmailBodyTemplatePath (note that subjects are provided directly in the configuration, and each email body is located in a separate resource or file). These templates can contain the following variables (all variable names are self-explanatory):

```
%LATE_DELIVERABLE_ID%
%PROJECT_NAME%
%PROJECT_VERSION%
%PROJECT_ID%
%PHASE_NAME%        (e.g. "Screening" or "Review")
%DELIVERABLE_NAME% (e.g. "Screening Scorecard" or "Review Scorecard")
```

```
%DEADLINE%          (the real deadline,
                     format depends on “timestampFormat” parameter)
%DELAY%             (e.g. "1 hour 12 minutes" or "2 days 0 hours 1 minute")
%COMPENSATED_DEADLINE% (is equal to the real deadline if no time should
             be compensated, format depends on “timestampFormat” parameter)
%COMPENSATED_AND_REAL_DEADLINES_DIFFER% (can be “true” or “false”)
%EXPLANATION%              (is empty if not yet provided by the user)
%EXPLANATION_CAN_BE_SENT% (can be “true” or “false”; is “true” only if
                           explanation has not been sent yet and
                           explanation deadline is not yet reached)
%EXPLANATION_DEADLINE%     (the deadline for sending an explanation,
                           format depends on “timestampFormat” parameter)
```

1.3.3.2 Templates of notifications for managers that have explained, but not responded late deliverables

These templates correspond to the fields emailSubjectTemplateText and emailBodyTemplatePath of NotRespondedLateDeliverablesNotifier (note that subject is provided directly in the configuration, and email body is located in a separate resource or file). These templates can contain the following variables (all variable names are self-explanatory):

```
%USER_ID%       (the ID of the PM user to which notification is sent)
%RECORDS_NUM%   (the number of explained but not responded late
                 deliverables for this PM user)
All fields below can be used inside of “RECORDS” loop:
  %LATE_DELIVERABLE_ID%
  %LATE_DELIVERABLE_TYPE% ("Missed Deadline" or "Rejected Final Fix")
  %PROJECT_NAME%
  %PROJECT_VERSION%
  %PROJECT_ID%
  %PHASE_NAME%        (e.g. "Screening" or "Review")
  %DELIVERABLE_NAME% (e.g. "Screening Scorecard" or "Review Scorecard")
  %DEADLINE%          (the real deadline,
                       format depends on “timestampFormat” parameter)
  %DELAY%           (e.g. "1 hour 12 minutes" or "2 days 0 hours 1 minute")
  %COMPENSATED_DEADLINE% (is equal to the real deadline if no time should
             be compensated, format depends on “timestampFormat” parameter)
  %COMPENSATED_AND_REAL_DEADLINES_DIFFER% (can be “true” or “false”)
  %LATE_MEMBER_HANDLE%
```

Note that for Rejected Final Fix notification emails DEADLINE, DELAY, COMPENSATED_DEADLINE and COMPENSATED_AND_REAL_DEADLINES_DIFFER parameters are always not specified.

1.4 Component Class Overview

Configurable [interface]

This interface should be extended by interfaces and implemented by classes that can be configured from Configuration API object. It's assumed that configure() method defined in this interface will be called just once for each instance.

EmailSendingUtility

This is a helper utility class that is used by LateDeliverableProcessorImpl and NotRespondedLateDeliverablesNotifier for sending warning email messages to the users who have late deliverables and the managers who need to provide responses for late deliverables respectively. This class supports constructing email message subjects and bodies from templates. It uses Document Generator component for this. This class uses Email Engine component to perform the actual sending of email messages.

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.

Changes in 1.3:

- Added type:LateDeliverableType property.

LateDeliverableProcessor [interface]

This interface represents a late deliverable processor. It provides a method for processing a single late deliverables. The actual actions to be performed depend on the implementation. This interface extends Configurable interface to support configuration via Configuration API component.

LateDeliverableProcessorImpl

This class is an implementation of LateDeliverableProcessor that uses pluggable ResourceManager and UserRetrieval instances to retrieve an additional information about the user who has a late deliverable, and DB Connection Factory component to perform auditing of all late deliverables and last sent notifications in the database. To send warning email messages to the users this class uses EmailSendingUtility. This class performs the logging of errors and debug information using Logging Wrapper component.

Changes in 1.3:

- Added support of Rejected Final Fix late deliverable type.

LateDeliverablesRetriever [interface]

This interface represents a late deliverables retriever. It provides a single method for retrieving all late deliverables of the specified types. This interface extends Configurable interface to support configuration via Configuration API component.

Changes in 1.3:

- Supporting retrieval of late deliverables of requested types only.

LateDeliverablesRetrieverImpl

This class is an implementation of LateDeliverablesRetriever that uses pluggable ProjectManager, PhaseManager and DeliverableManager instances to access data in persistence. It looks for all active projects with "Track Late Deliverables" property set to "true", and then retrieves all incomplete deliverables for all late phases or open Final Fix phases that go directly after Final Review. This class performs the logging of errors and debug information using Logging Wrapper component.

Changes in 1.3:

- Supporting retrieval of late deliverables of requested types only.

- Added support of Rejected Final Fix late deliverables.

LateDeliverablesTracker

This class provides a programmatic API for tracking late deliverables of the configured types. It provides just a single execute() method that uses pluggable LateDeliverablesRetriever and LateDeliverableProcessor instances to find and process all late deliverables of the configured types. This class performs the logging of errors and debug information using Logging Wrapper component.

Changes in 1.3:

- Added support of configurable list of late deliverable types to be tracked.

LateDeliverableType [enum]

This is an enumeration for late deliverable types. Currently two late deliverable types are supported by this component: Missed Deadline and Rejected Final Fix.

LateDeliverablesTrackingJobRunner

This class is an implementation of ScheduledJobRunner that aggregates an instance of LateDeliverablesTracker and can be used for scheduling the late deliverables tracking with use of Job Scheduling and Job Processor components. This job runner doesn't allow two jobs to be executed at the same time, thus if the previous job is not yet finished, a new one is not started.

LateDeliverablesTrackingUtility

This is the main class of the standalone command line application that performs periodical late deliverables tracking. It uses LateDeliverablesTrackingJobRunner and NotRespondedLateDeliverablesNotificationJobRunner, and schedules their repetitive execution with use of Job Scheduling and Job Processor components. This utility reads a configuration from a file using Configuration Persistence and Configuration API components. LateDeliverablesTrackingUtility performs the logging of errors and debug information using Logging Wrapper.

Changes in 1.2:

- "interval" switch was renamed to "trackingInterval".

- Added support for "notificationInterval" switch.

- Added support for scheduling not responded late deliverables notification job.

NotRespondedLateDeliverablesNotificationJobRunner

This class is an implementation of ScheduledJobRunner that aggregates an instance of NotRespondedLateDeliverablesNotifier and can be used for scheduling the sending of notifications about all explained but not responded late deliverables to the managers with use of Job Scheduling and Job Processor components. This job runner doesn't allow two jobs to be executed at the same time, thus if the previous job is not yet finished, a new one is not started.

NotRespondedLateDeliverablesNotifier

This class provides a programmatic API for sending email notification about explained but not responded late deliverables to the managers. It provides just a single execute() method that sends notifications for all such late deliverables. Note that a single email message is sent to each manager, and this email message contains information about all late deliverables to be responded by this manager. This class performs the logging of errors and debug information using Logging Wrapper component.

Changes in 1.3:

- Added support of "LATE_DELIVERABLE_TYPE" email template field.

NotRespondedLateDeliverablesNotifier.LateDeliverableDetails

This is an inner class of NotRespondedLateDeliverablesNotifier. It is a container for information about a single explained but not responded 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.

Changes in 1.3:

- Added lateDeliverableTypeName property.

1.5 Component Exception Definitions

EmailSendingException

This exception is thrown by LateDeliverableProcessorImpl and EmailSendingUtility when some error occurs while sending a notification email message.

Changes in 1.2:

- Moved from com.topcoder.management.deliverable.latetracker.processors to

com.topcoder.management.deliverable.latetracker package.

LateDeliverablesProcessingException
This exception is thrown by LateDeliverablesTracker and implementations of LateDeliverableProcessor when some error occurs while processing a late deliverable. Also this exception is used as a base class for other implementation specific custom exceptions.

LateDeliverablesRetrievalException
This exception is thrown by LateDeliverablesTracker and implementations of LateDeliverablesRetriever when some error occurs while retrieving a list of late deliverables.

LateDeliverablesTrackerConfigurationException
This exception is thrown by LateDeliverablesTracker, NotRespondedLateDeliverablesNotifier and implementations of Configurable when some error occurs while initializing an instance using the given configuration.

LateDeliverablesTrackingException
This exception is a base class for all other custom checked exceptions defined in this component. It is never thrown directly, subclasses are used instead.

NotRespondedLateDeliverablesNotificationException
This exception is thrown by NotRespondedLateDeliverablesNotifier when some error occurs while retrieving not responded late deliverables or sending email notifications to managers.

1.6 Thread Safety

This component is not thread safe.

LateDeliverablesTrackingUtility is immutable and thread safe. But it's not safe to execute multiple instances of LateDeliverablesTrackingUtility command line application (configured to use the same persistence) at a time.

Implementations of LateDeliverableProcessor, LateDeliverablesRetriever and Configurable are not required to be thread safe.

LateDeliverablesTracker is immutable, but not thread safe since it uses LateDeliverablesRetriever and LateDeliverableProcessor instances that are not guaranteed to be thread safe.

NotRespondedLateDeliverablesNotifier is immutable, but not thread safe since it uses ProjectManager, ResourceManager and DeliverablePersistence instances that are not guaranteed to be thread safe.

LateDeliverable and LateDeliverableDetails are mutable and not thread safe entities.

LateDeliverableType is thread safe enumeration class.

LateDeliverablesTrackingJobRunner is mutable, but it uses additional synchronization when accessing any mutable attribute (except lateDeliverablesTracker and log attributes that are assumed to be immutable after initialization). It's assumed that configure() method will be called just once right after instantiation or static setConfig() method will be called instead, before calling any business methods. LateDeliverablesTrackingJobRunner uses a not thread safe LateDeliverablesTracker instance, but it guarantees that it will be accessed from one thread only at a time (but not allowing to run two simultaneous jobs).

NotRespondedLateDeliverablesNotificationJobRunner is mutable, but it uses additional synchronization when accessing any mutable attribute (except notRespondedLateDeliverableNotifier and log attributes that are assumed to be immutable after initialization). It's assumed that configure() method will be called just once right after instantiation or static setConfig() method will be called instead, before calling any business methods. This class uses a not thread safe NotRespondedLateDeliverablesNotifier instance, but it guarantees that it will be accessed from one thread only at a time (but not allowing to run two simultaneous jobs).

LateDeliverableProcessorImpl is not thread safe since it uses ResourceManager instance that is not thread safe. It's assumed that configure() method will be called just once right after instantiation, before calling any business methods. LateDeliverableProcessorImpl uses transactions when inserting or updating data in persistence.

LateDeliverablesRetrieverImpl is not thread safe since it uses ProjectManager, PhaseManager, DeliverableManager and ResourceManager instances that are not thread safe. It's assumed that configure() method will be called just once right after instantiation, before calling any business methods.

EmailSendingUtility is immutable and thread safe. It uses thread safe EmailEngine class.

Thread safety of this component was not changed in the version 1.3.

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.

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

Email Engine 3.2 – is used for sending email messages.

Document Generator 3.1.1 – is used for generating text of email messages from templates.

Command Line Utility 1.0 – is used for parsing command line arguments.

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

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.

Search Builder 1.3.1 – defines Filter and OrFilter entities used in this component.

Job Scheduling 3.2 – is used for loading job schedule from ConfigurationObject, defines ScheduledJobRunner interface implemented in this component.

Job Processor 3.0.1 – is used for executing late deliverables tracking job periodically.

User Project Data Store 1.0.1 – defines UserRetrieval interface and ExternalUser entity used in this component.

Project Management 1.0.1 – defines ProjectManager, ProjectFilterUtility and Project entity used in this component.

Phase Management 1.0.4 – defines PhaseManager interface used in this component.

Deliverable Management 1.1.1 – defines PersistenceDeliverableManager, DeliverableFilterBuilder, DeliverableChecker and Deliverable entity used in this component.

Project Phases 2.0.1 – defines phase specific entities used in this component.

Resource Management 1.1.1 – defines ResourceManager interface and Resource entity used in this component.

Late Deliverables Management 1.0.6 – is used for retrieving explained but not responded late deliverables.

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.latetracker
com.topcoder.management.deliverable.latetracker.notification
com.topcoder.management.deliverable.latetracker.processors
com.topcoder.management.deliverable.latetracker.retrievers
com.topcoder.management.deliverable.latetracker.utility

3.2 Configuration Parameters

3.2.1 Configuration of LateDeliverablesTracker

The following table describes the structure of ConfigurationObject passed to the constructor of LateDeliverablesTracker 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.
<objectFactoryConfig>	This section contains configuration of Object Factory used by this class for creating pluggable object instances.	ConfigurationObject. Required.
lateDeliverablesRetrieverKey	The Object Factory key that is used for creating an instance of LateDeliverablesRetriever to be used by this class.	String. Not empty. Required.
lateDeliverablesRetriever Config	The configuration of LateDeliverablesRetriever to be used by this class.	ConfigurationObject. Required.
lateDeliverableProcessorKey	The Object Factory key that is used for creating an instance of LateDeliverableProcessor to be used by this class.	String. Not empty. Required.
lateDeliverableProcessor Config	The configuration of LateDeliverableProcessor to be used by this class.	ConfigurationObject. Required.
lateDeliverableTypes	The types of late deliverables to be tracked. Valid values are “Missed Deadline” and “Rejected Final Fix”. If not specified, late deliverables of all types are tracked.	String[]. Not empty, cannot contain duplicate elements. Optional.

3.2.2 Configuration of LateDeliverablesTrackingJobRunner

The following table describes the structure of ConfigurationObject passed to the configure() method of LateDeliverablesTrackingJobRunner 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.

Additionally this configuration object should contain all parameters of LateDeliverablesTracker. See section 3.2.1 for details.

3.2.3 Configuration of LateDeliverablesRetrieverImpl

The following table describes the structure of ConfigurationObject passed to the configure() method of LateDeliverablesRetrieverImpl 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.
missedDeadlineTracking DeliverableIds	The comma separated list of deliverable IDs for which Missed Deadline tracking must be performed. Each element in the list must be a positive long integer.	String. Not empty. Required.
<objectFactoryConfig>	This section contains configuration of Object Factory used by this class for creating pluggable object instances.	ConfigurationObject. Required.
projectManagerKey	The Object Factory key that is used for creating an instance of ProjectManager to be used by this class.	String. Not empty. Required.
phaseManagerKey	The Object Factory key that is used for creating an instance of PhaseManager to be used by this class.	String. Not empty. Required.
resourceManagerKey	The Object Factory key that is used for creating an instance of ResourceManager to be used by this class.	String. Not empty. Required.
deliverablePersistenceKey	The Object Factory key that is used for creating an instance of DeliverablePersistence to be used by this class.	String. Not empty. Required.
<deliverableCheckerXXX>	Here XXX is any substring, e.g. “1”, “2”, etc. This section contains details for a single deliverable checker to be used when creating PersistenceDeliverableManager instance.	ConfigurationObject. Multiple. At least one is required.
<deliverableCheckerXXX>. deliverableName	The name of the deliverable for this deliverable checker.	String. Not empty. Required.

<deliverableCheckerXXX>.deliverableCheckerKey	The Object Factory key that is used for creating an instance of DeliverableChecker to be associated with the specified deliverable name.	String. Not empty. Required.
searchBundleManagerNamespace	The namespace used when creating a SearchBundleManager instance. This SearchBundleManager instance is next used for creating PersistenceDeliverableManager.	String. Not empty. Required.
maxDurationOfPhaseWithCompensatedDeadline	The maximum duration of the phase in milliseconds (not inclusive) for which compensated deadline should be calculated. Default is “86400000” (24 hours).	String representation of not negative long integer. Optional.

3.2.4 Configuration of LateDeliverableProcessorImpl

The following table describes the structure of ConfigurationObject passed to the configure() method of LateDeliverableProcessorImpl 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.
connectionName	The name of connection in database connection factory. When not provided, default connection will be retrieved.	String. Not empty. Optional.
<objectFactoryConfig>	This section contains configuration of Object Factory used by this class for creating pluggable object instances.	ConfigurationObject. Required.
<dbConnectionFactoryConfig>	The configuration to be used for creating DBConnectionFactoryImpl instance.	ConfigurationObject. Required.
missedDeadlineNotificationDeliverableIds	The comma separated list of deliverable IDs for which sending of notifications must be performed. Is used for late deliverables of Missed Deadline type. Each element in the list must be a positive long integer. If this parameter is not specified or empty, warning emails are not sent for all Missed Deadline deliverables.	String. Optional.
missedDeadlineEmailSubjectForDeliverableX	Here “X” is a positive integer that represents the deliverable ID. The email subject template text to be used for late deliverables with the specified ID. Is used for late deliverables of Missed Deadline type.	String. Not empty. Multiple. Optional.

missedDeadlineEmail BodyForDeliverableX	Here "X" is a positive integer that represents the deliverable ID. The email body template path (resource path or file path) to be used for late deliverables with the specified ID. Is used for late deliverables of Missed Deadline type.	String. Not empty. Multiple. Optional.
defaultMissedDeadlineEmail SubjectTemplateText	The default email subject template text to be used for all deliverable IDs not configured with "missedDeadlineEmailSubjectForDeliverableX" parameters. Is used for late deliverables of Missed Deadline type.	String. Required.
defaultMissedDeadlineEmail BodyTemplatePath	The default email body template path (resource path or file path) to be used for all deliverable IDs not configured with "missedDeadlineEmailBodyForDeliverableX" parameters. Is used for late deliverables of Missed Deadline type.	String. Not empty. Required.
rejectedFinalFixEmail SubjectTemplateText	The email subject template text to be used for sending notifications about Rejected Final Fix late deliverables.	String. Required.
rejectedFinalFixEmail BodyTemplatePath	The email body template path (resource path or file path) to be used for sending notifications about Rejected Final Fix late deliverables.	String. Not empty. Required.
emailSender	The address of the email sender to be used.	String. Not empty. Required.
resourceManagerKey	The Object Factory key that is used for creating an instance of ResourceManager to be used by this class.	String. Not empty. Required.
userRetrievalKey	The Object Factory key that is used for creating an instance of UserRetrieval service to be used by this class.	String. Not empty. Required.
timestampFormat	The timestamp format to be used for formatting timestamps in the email message. See JDK docs of SimpleDateFormat class for details. Default is "yyyy-MM-dd HH:mm".	String. Not empty. Optional.
notificationInterval	The interval in seconds between sending notifications to the user about the same late deliverable. Is used for late deliverables of Missed Deadline type only. If not specified, notifications are not sent repeatedly.	String representation of positive integer. Optional.
explanationDeadlineIntervalIn Hours	The time interval between the late record creation date and the explanation deadline (in hours). Default is "24".	String representation of positive integer. Optional.

lateDeliverableTypeIds	The mapping from LateDeliverableType enum value to its corresponding record ID in the database. Must be in format “Missed Deadline=XXX,Rejected Final Fix=YYY”, where XXX and YYY must be replaced with positive integer IDs.	String. Not empty. Required.
sendRejectedFinalFix Notifications	The value indicating whether notifications must be sent for Rejected Final Fix late deliverables. Default is “true”.	“true” or “false”. Optional.

3.2.5 Configuration of LateDeliverablesTrackingUtility

The following table describes the structure of ConfigurationObject used in the main() method of LateDeliverablesTrackingUtility class (angle brackets are used for identifying child configuration objects). This ConfigurationObject is 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.
<schedulerConfig>	The configuration used for creating ConfigurationObjectScheduler instance.	ConfigurationObject. Required.
trackingJobName	The name of the job from scheduler configuration that corresponds to LateDeliverablesTrackingJobRunner. Note that the type of this job must be “JOB_TYPE_JAVA_CLASS”, and run command – equal to the full class name of LateDeliverablesTrackingJobRunner.	String. Not empty. Required.
<trackingJobConfig>	The configuration used for initializing LateDeliverablesTrackingJobRunner. Please see section 3.2.2 for details.	ConfigurationObject. Required.
notificationJobName	The name of the job from scheduler configuration that corresponds to NotRespondedLateDeliverablesNotificationJobRunner. Note that the type of this job must be “JOB_TYPE_JAVA_CLASS”, and run command – equal to the full class name of NotRespondedLateDeliverablesNotificationJobRunner.	String. Not empty. Required.
<notificationJobConfig>	The configuration used for initializing NotRespondedLateDeliverablesNotificationJobRunner. Please see section 3.2.8 for details.	ConfigurationObject. Required.

3.2.6 LateDeliverablesTrackingUtility command line parameters

The following table describes the command line switches and arguments that are supported by LateDeliverablesTrackingUtility standalone application.

Switch and arguments	Description
-c <file_name>	Optional. Provides the name of the configuration file for this command line application. This file is read with use of Configuration Persistence component. Default is “com/topcoder/management/deliverable/latetracker/utility/LateDeliverablesTrackingUtility.properties”.
-ns <namespace>	Optional. The namespace in the specified configuration file that contains configuration for this command line application. Default is “com.topcoder.management.deliverable.latetracker.utility.LateDeliverablesTrackingUtility”.
-trackingInterval <interval_in_sec>	Optional. The interval in seconds between checks of projects for late deliverables. If not specified, the value from the scheduler configuration is used (see section 3.2.5 of CS).
-notificationInterval <interval_in_sec>	Optional. The interval in seconds between sending notifications about explained but not responded late deliverables to the managers. If not specified, the value from the scheduler configuration is used (see section 3.2.5 of CS).
-guardFile <file_path>	Required. The path to guard file which should be used to signal to Late Deliverables Tracker that it has to stop.
-background [true|false]	Required. The flag indicating whether the tracker is going to run in background thread or not.
-help -? -h	When one of the specified switches is provided, the application prints out the usage string to the standard output and terminates immediately.

3.2.7 Configuration of NotRespondedLateDeliverablesNotifier

The following table describes the structure of ConfigurationObject passed to the constructor of NotRespondedLateDeliverablesNotifier 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.
<objectFactoryConfig>	This section contains configuration of Object Factory used by this class for creating pluggable object instances.	ConfigurationObject. Required.
lateDeliverableManagerKey	The Object Factory key that is used for creating an instance of LateDeliverableManager to be used by this class.	String. Not empty. Required.
managerResourceRoleIds	The list of IDs of resource roles that represent managers to which notifications about not responded late deliverables are sent.	String[]. Not empty. Each element must be a valid positive integer. Required.

resourceManagerKey	The Object Factory key that is used for creating an instance of ResourceManager to be used by this class.	String. Not empty. Required.
userRetrievalKey	The Object Factory key that is used for creating an instance of UserRetrieval service to be used by this class.	String. Not empty. Required.
emailSubjectTemplateText	The email subject template text to be used for constructing messages to be sent to the managers.	String. Required.
emailBodyTemplatePath	The email body template path (resource path or file path) to be used for constructing messages to be sent to the managers.	String. Not empty. Required.
emailSender	The address of the email sender to be used.	String. Not empty. Required.
timestampFormat	The timestamp format to be used for formatting timestamps in the email message. See JDK docs of SimpleDateFormat class for details. Default is “yyyy-MM-dd HH:mm”.	String. Not empty. Optional.
projectManagerKey	The Object Factory key that is used for creating an instance of ProjectManager to be used by this class.	String. Not empty. Required.
phasePersistenceKey	The Object Factory key that is used for creating an instance of PhasePersistence to be used by this class.	String. Not empty. Required.
deliverablePersistenceKey	The Object Factory key that is used for creating an instance of DeliverablePersistence to be used by this class.	String. Not empty. Required.

3.2.8 Configuration of NotRespondedLateDeliverablesNotificationJobRunner

The following table describes the structure of ConfigurationObject passed to the configure() method of NotRespondedLateDeliverablesNotificationJobRunner 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.

Additionally this configuration object should contain all parameters of NotRespondedLateDeliverablesNotifier. See section 3.2.7 for details.

3.3 Dependencies Configuration

Please see docs of dependency components to configure them properly.

4. Usage Notes

4.1 Required steps to test the component

- Extract the component distribution.

- Follow Dependencies Configuration.
- Setup Informix database:

  a. Run test_files/all.sql.

  b. Update test_files/DB_Factory.xml, LateDeliverableProcessorImpl.xml, LateDeliverablesTracker.xml, LateDeliverablesTrackingUtility.xml

  and invalid_config/*xml if needed.
- Start test_files/DevNullSmtp.jar.
- 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 API usage

```
// Prepare configuration for LateDeliverablesRetriever
ConfigurationObject lateDeliverablesRetrieverConfig =
    getConfigurationObject("config/LateDeliverablesRetrieverImpl.xml",
        LateDeliverablesRetrieverImpl.class.getName());

// Prepare configuration for LateDeliverableProcessor
ConfigurationObject lateDeliverableProcessorConfig =
    getConfigurationObject("config/LateDeliverableProcessorImpl.xml",
        LateDeliverableProcessorImpl.class.getName());

// Create an instance of LateDeliverablesRetrieverImpl and configure it
LateDeliverablesRetriever lateDeliverablesRetriever = new LateDeliverablesRetrieverImpl();
lateDeliverablesRetriever.configure(lateDeliverablesRetrieverConfig);

// Create an instance of LateDeliverableProcessorImpl and configure it
LateDeliverableProcessor lateDeliverableProcessor = new LateDeliverableProcessorImpl();
lateDeliverableProcessor.configure(lateDeliverableProcessorConfig);

// Get logger
Log log = LogFactory.getLog("my_logger");

// Create LateDeliverablesTracker
Set<LateDeliverableType> trackedLateDeliverableTypes = new EnumSet<LateDeliverableType>();
trackedLateDeliverableTypes.add(LateDeliverableType.MISSED_DEADLINE);
trackedLateDeliverableTypes.add(LateDeliverableType.REJECTED_FINAL_FIX);
LateDeliverablesTracker lateDeliverablesTracker = new LateDeliverablesTracker(
    lateDeliverablesRetriever, lateDeliverableProcessor, log, trackedLateDeliverableTypes);

// Track for late deliverables
lateDeliverablesTracker.execute();

...

// Prepare configuration for NotRespondedLateDeliverablesNotifier
ConfigurationObject notifierConfig = ...

// Create an instance of NotRespondedLateDeliverablesNotifier
NotRespondedLateDeliverablesNotifier notifier =
    new NotRespondedLateDeliverablesNotifier(notifierConfig);

// Send notifications for explained but not responded late deliverables
notifier.execute();
```

4.3.2 Usage of command line utility

This command line can be used to print out the usage string:

```
java com.topcoder.management.deliverable.latetracker.utility.LateDeliverablesTrackingUtility
-help
```

If configuration for the utility is stored in the default namespace of the default configuration file, then the application can be executed in background with the following arguments:

```
java com.topcoder.management.deliverable.latetracker.utility.LateDeliverablesTrackingUtility
–guardFile guard.tmp –background true
```

To use the custom configuration file the user can provide "-c" switch:

```
java com.topcoder.management.deliverable.latetracker.utility.LateDeliverablesTrackingUtility
–c custom_config.properties –guardFile guard.tmp –background true
```

The user can specify custom import files utility configuration file name and namespace:

```
java com.topcoder.management.deliverable.latetracker.utility.LateDeliverablesTrackingUtility
–c custom_config.properties –ns custom_namespace –guardFile guard.tmp –background true
```

The user can specify interval between late deliverable checks and interval between sending PM notifications in the command line (in this example deliverables will be checked every 5 minutes, and notifications will be sent every hour):

```
java com.topcoder.management.deliverable.latetracker.utility.LateDeliverablesTrackingUtility
–trackingInterval 300 –notificationInterval 3600 –guardFile guard.tmp –background true
```

4.3.3 Sample LateDeliverablesTrackingUtility configuration file

The file provided in this section is not a file that is specified as the command line argument. Instead the command line should contain the name of the properties file passed to ConfigurationFileManager. And this properties file should contain the link to the actual LateDeliverablesTrackingUtility configuration file, sample of which is provided in this section.

```
<?xml version="1.0"?>
<CMConfig>
  <Config name=
   "com.topcoder.management.deliverable.latetracker.utility.LateDeliverablesTrackingUtility">
    <Property name="loggerName">
      <Value>myLogger</Value>
    </Property>
    <Property name="schedulerConfig">
      <Property name="lateDeliverablesTrackingJob">
        <Property name="StartDate">
          <Value>Jan 1, 2010 00:00:00 AM</Value>
        </Property>
        <Property name="StartTime">
          <Value>1000</Value>
        </Property>
        <Property name="EndDate">
          <Value>Jan 1, 2100 00:00:00 AM</Value>
        </Property>
        <Property name="JobType">
          <Value>JOB_TYPE_JAVA_CLASS</Value>
        </Property>
        <Property name="JobCommand">
          <Value>
            com.topcoder.management.deliverable.latetracker.LateDeliverablesTrackingJobRunner
          </Value>
        </Property>
        <Property name="Active">
          <Value>True</Value>
        </Property>
        <Property name="ModificationDate">
          <Value>Sep 13, 2010 05:00:00 AM</Value>
        </Property>
        <Property name="Recurrence">
          <Value>1</Value>
        </Property>
        <Property name="Interval">
          <Property name="Value">
            <Value>20</Value>
          </Property>
          <Property name="Unit">
            <Property name="Type">
```

```
              <Value>com.topcoder.util.scheduler.scheduling.Second</Value>
            </Property>
          </Property>
        </Property>
      </Property>
      <Property name="notRespondedLateDeliverablesNotificationJob">
        <Property name="StartDate">
          <Value>Jan 1, 2010 00:00:00 AM</Value>
        </Property>
        <Property name="StartTime">
          <Value>1000</Value>
        </Property>
        <Property name="EndDate">
          <Value>Jan 1, 2100 00:00:00 AM</Value>
        </Property>
        <Property name="JobType">
          <Value>JOB_TYPE_JAVA_CLASS</Value>
        </Property>
        <Property name="JobCommand">
          <Value>
com.topcoder.management.deliverable.latetracker.notification.NotRespondedLateDeliverablesNoti
ficationJobRunner
          </Value>
        </Property>
        <Property name="Active">
          <Value>True</Value>
        </Property>
        <Property name="ModificationDate">
          <Value>Mar 10, 2011 09:00:00 AM</Value>
        </Property>
        <Property name="Recurrence">
          <Value>1</Value>
        </Property>
        <Property name="Interval">
          <Property name="Value">
            <Value>21600</Value>
          </Property>
          <Property name="Unit">
            <Property name="Type">
              <Value>com.topcoder.util.scheduler.scheduling.Second</Value>
            </Property>
          </Property>
        </Property>
      </Property>
    </Property>
    <Property name="trackingJobName">
      <Value>lateDeliverablesTrackingJob</Value>
    </Property>

    <Property name="trackingJobConfig">
      <Property name="loggerName">
        <Value>myLogger</Value>
      </Property>

      <Property name="objectFactoryConfig">
        <property name="lateDeliverablesRetriever">
          <property name="type">
            <value>
     com.topcoder.management.deliverable.latetracker.retrievers.LateDeliverablesRetrieverImpl
            </value>
          </property>
        </property>
        <property name="lateDeliverableProcessor">
          <property name="type">
            <value>
      com.topcoder.management.deliverable.latetracker.processors.LateDeliverableProcessorImpl
            </value>
          </property>
        </property>
      </Property>
      <Property name="lateDeliverableProcessorKey">
```

```
  <Value>lateDeliverableProcessor</Value>
</Property>

<Property name="lateDeliverablesRetrieverKey">
  <Value>lateDeliverablesRetriever</Value>
</Property>

<Property name="lateDeliverablesRetrieverConfig">
  <Property name="loggerName">
    <Value>myLogger</Value>
  </Property>
  <Property name="resourceManagerKey">
    <Value>resourceManager</Value>
  </Property>
  <Property name="missedDeadlineTrackingDeliverableIds">
    <Value>3,4</Value>
  </Property>
  <Property name="objectFactoryConfig">
    <property name="resourceManager">
      <property name="type">
        <value>
          com.topcoder.management.resource.persistence.PersistenceResourceManager
        </value>
      </property>
      <Property name="params">
        <Property name="param1">
          <Property name="name">
            <Value>ResourcePersistence</Value>
          </Property>
        </Property>
        <Property name="param2">
          <Property name="name">
            <Value>SearchBundleManager</Value>
          </Property>
        </Property>
      </Property>
    </property>
    <property name="ResourcePersistence">
      <property name="type">
        <value>
          com.topcoder.management.resource.persistence.sql.SqlResourcePersistence
        </value>
      </property>
      <Property name="params">
        <Property name="param1">
          <Property name="name">
            <Value>DBConnectionFactory</Value>
          </Property>
        </Property>
      </Property>
    </property>
    <property name="SearchBundleManager">
      <property name="type">
        <value>com.topcoder.search.builder.SearchBundleManager</value>
      </property>
      <Property name="params">
        <Property name="param1">
          <Property name="type">
            <Value>String</Value>
          </Property>
          <Property name="value">
            <Value>com.topcoder.search.builder.SearchBundleManager</Value>
          </Property>
        </Property>
      </Property>
    </property>
    <property name="projectManager">
      <property name="type">
        <value>com.topcoder.management.project.ProjectManagerImpl</value>
      </property>
    </property>
```

```
  <property name="phaseManager">
    <property name="type">
      <value>com.topcoder.management.phase.DefaultPhaseManager</value>
    </property>
    <Property name="params">
      <Property name="param1">
        <Property name="type">
          <Value>String</Value>
        </Property>
        <Property name="value">
          <Value>com.topcoder.management.phase.DefaultPhaseManager</Value>
        </Property>
      </Property>
    </Property>
  </property>
  <property name="deliverablePersistence">
    <property name="type">
      <value>
        com.topcoder.management.deliverable.persistence.sql.SqlDeliverablePersistence
      </value>
    </property>
    <Property name="params">
      <Property name="param1">
        <Property name="name">
          <Value>DBConnectionFactory</Value>
        </Property>
      </Property>
    </Property>
  </property>
  <property name="DBConnectionFactory">
    <property name="type">
      <value>com.topcoder.db.connectionfactory.DBConnectionFactoryImpl</value>
    </property>
    <Property name="params">
      <Property name="param1">
        <Property name="type">
          <Value>String</Value>
        </Property>
        <Property name="value">
          <Value>com.topcoder.db.connectionfactory.DBConnectionFactoryImpl</Value>
        </Property>
      </Property>
    </Property>
  </property>
  <property name="screeningDeliverableChecker">
    <property name="type">
      <value>
        com.topcoder.management.deliverable.latetracker.MockDeliverableChecker
      </value>
    </property>
  </property>
  <property name="reviewDeliverableChecker">
    <property name="type">
      <value>
        com.topcoder.management.deliverable.latetracker.MockDeliverableChecker
      </value>
    </property>
  </property>
</Property>
<Property name="projectManagerKey">
  <Value>projectManager</Value>
</Property>
<Property name="phaseManagerKey">
  <Value>phaseManager</Value>
</Property>
<Property name="deliverablePersistenceKey">
  <Value>deliverablePersistence</Value>
</Property>
<Property name="deliverableChecker1">
  <Property name="deliverableName">
    <Value>Screening Scorecard</Value>
```

```
    </Property>
    <Property name="deliverableCheckerKey">
      <Value>screeningDeliverableChecker</Value>
    </Property>
  </Property>
  <Property name="deliverableChecker2">
    <Property name="deliverableName">
      <Value>Review Scorecard</Value>
    </Property>
    <Property name="deliverableCheckerKey">
      <Value>reviewDeliverableChecker</Value>
    </Property>
  </Property>
  <Property name="searchBundleManagerNamespace">
    <Value>com.topcoder.search.builder.SearchBundleManager</Value>
  </Property>
</Property>
<Property name="lateDeliverableProcessorConfig">
  <Property name="loggerName">
    <Value>myLogger</Value>
  </Property>
  <Property name="connectionName">
    <Value>informix_connection</Value>
  </Property>
  <Property name="objectFactoryConfig">
    <property name="resourceManager">
      <property name="type">
        <value>
          com.topcoder.management.resource.persistence.PersistenceResourceManager
        </value>
      </property>
      <Property name="params">
        <Property name="param1">
          <Property name="name">
            <Value>ResourcePersistence</Value>
          </Property>
        </Property>
        <Property name="param2">
          <Property name="name">
            <Value>SearchBundleManager</Value>
          </Property>
        </Property>
      </Property>
    </property>
    <property name="ResourcePersistence">
      <property name="type">
        <value>
          com.topcoder.management.resource.persistence.sql.SqlResourcePersistence
        </value>
      </property>
      <Property name="params">
        <Property name="param1">
          <Property name="name">
            <Value>DBConnectionFactory</Value>
          </Property>
        </Property>
      </Property>
    </property>
    <property name="DBConnectionFactory">
      <property name="type">
        <value>com.topcoder.db.connectionfactory.DBConnectionFactoryImpl</value>
      </property>
      <Property name="params">
        <Property name="param1">
          <Property name="type">
            <Value>String</Value>
          </Property>
          <Property name="value">
            <Value>com.topcoder.db.connectionfactory.DBConnectionFactoryImpl</Value>
          </Property>
        </Property>
```

```
    </Property>
  </property>
  <property name="SearchBundleManager">
    <property name="type">
      <value>com.topcoder.search.builder.SearchBundleManager</value>
    </property>
    <Property name="params">
      <Property name="param1">
        <Property name="type">
          <Value>String</Value>
        </Property>
        <Property name="value">
          <Value>com.topcoder.search.builder.SearchBundleManager</Value>
        </Property>
      </Property>
    </Property>
  </property>
  <property name="userRetrieval">
    <property name="type">
      <value>com.cronos.onlinereview.external.impl.DBUserRetrieval</value>
    </property>
    <Property name="params">
      <Property name="param1">
        <Property name="type">
          <Value>String</Value>
        </Property>
        <Property name="value">
          <Value>com.topcoder.db.connectionfactory.DBConnectionFactoryImpl</Value>
        </Property>
      </Property>
    </Property>
  </property>
</Property>
<Property name="dbConnectionFactoryConfig">
  <Property name="com.topcoder.db.connectionfactory.DBConnectionFactoryImpl">
    <Property name="connections">
      <Property name="informix_connection">
        <Property name="producer">
          <Value>
            com.topcoder.db.connectionfactory.producers.JDBCConnectionProducer
          </Value>
        </Property>
        <Property name="parameters">
          <Property name="jdbc_driver">
            <Value>com.informix.jdbc.IfxDriver</Value>
          </Property>
          <Property name="jdbc_url">
            <Value>
              jdbc:informix-sqli://192.168.1.3:9090/tcs:informixserver=topcoder
            </Value>
          </Property>
          <Property name="user">
            <Value>informix</Value>
          </Property>
          <Property name="password">
            <Value>123456</Value>
          </Property>
        </Property>
      </Property>
    </Property>
  </Property>
</Property>
<Property name="missedDeadlineNotificationDeliverableIds">
  <Value>4</Value>
</Property>
<Property name="defaultMissedDeadlineEmailSubjectTemplateText">
  <Value>
    WARNING\: You are late when providing a deliverable for %PROJECT_NAME%
  </Value>
</Property>
<Property name="defaultMissedDeadlineEmailBodyTemplatePath">
```

```
      <Value>test_files/warn_email_template.html</Value>
    </Property>
    <Property name="rejectedFinalFixEmailSubjectTemplateText">
      <Value>
        WARNING\: You need to explain why your Final Fix for %PROJECT_NAME% was rejected
      </Value>
    </Property>
    <Property name="rejectedFinalFixEmailBodyTemplatePath">
      <Value>test_files/rejected_ff_email_template.html</Value>
    </Property>
    <Property name="emailSender">
      <Value>service@topcoder.com</Value>
    </Property>
    <Property name="resourceManagerKey">
      <Value>resourceManager</Value>
    </Property>
    <Property name="userRetrievalKey">
      <Value>userRetrieval</Value>
    </Property>
    <Property name="timestampFormat">
      <Value>yyyy-MM-dd HH:mm:ss</Value>
    </Property>
    <Property name="notificationInterval">
      <Value>600</Value>
    </Property>
    <Property name="explanationDeadlineIntervalInHours">
      <Value>24</Value>
    </Property>
    <Property name="lateDeliverableTypeIds">
      <Value>Missed Deadline=1,Rejected Final Fix=2</Value>
    </Property>
    <Property name="sendRejectedFinalFixNotifications">
      <Value>true</Value>
    </Property>
  </Property>
  <Property name="lateDeliverableTypes">
    <Value>Missed Deadline</Value>
    <Value>Rejected Final Fix</Value>
  </Property>
</Property>

<Property name="notificationJobName">
  <Value>notRespondedLateDeliverablesNotificationJob</Value>
</Property>

<Property name="notificationJobConfig">
  <Property name="loggerName">
    <Value>myLogger</Value>
  </Property>

  <Property name="objectFactoryConfig">
    <property name="lateDeliverableManager">
      <property name="type">
        <value>
            com.topcoder.management.deliverable.late.impl.LateDeliverableManagerImpl
        </value>
      </property>
    </property>

    <!-- Put the rest of the Object Factory configuration here -->

  </Property>
  <Property name="lateDeliverableManagerKey">
    <Value>lateDeliverableManager</Value>
  </Property>
  <Property name="managerResourceRoleIds">
    <Value>13</Value>
    <Value>14</Value>
  </Property>
  <Property name="resourceManagerKey">
    <Value>resourceManager</Value>
```

```
      </Property>
      <Property name="userRetrievalKey">
        <Value>userRetrieval</Value>
      </Property>
      <Property name="emailSubjectTemplateText">
        <Value>
          WARNING\: You have explained late deliverable(s) to be responded
        </Value>
      </Property>
      <Property name="emailBodyTemplatePath">
        <Value>test_files/pm_notification_email_template.html</Value>
      </Property>
      <Property name="emailSender">
        <Value>service@topcoder.com</Value>
      </Property>
      <Property name="timestampFormat">
        <Value>yyyy-MM-dd HH:mm:ss</Value>
      </Property>
      <Property name="projectManagerKey">
        <Value>projectManager</Value>
      </Property>
      <Property name="phasePersistenceKey">
        <Value>phasePersistence</Value>
      </Property>
      <Property name="deliverablePersistenceKey">
        <Value>deliverablePersistence</Value>
      </Property>
    </Property>
  </Config>
</CMConfig>
```

4.3.4 Sample late deliverable warning email body template

warn_email_template.html

```
<p>
Deadline: %DEADLINE%<br/>
Your current delay is: %DELAY%<br/>
Contest link: <a
href="http://software.topcoder.com/review/actions/ViewProjectDetails.do?pid=%PROJECT_ID%">%PR
OJECT_NAME%</a><br/>
%if:COMPENSATED_AND_REAL_DEADLINES_DIFFER = 'true'%
<b>Note that the deadline was compensated due to the dependency phases having ended
prematurely.<br/>
The compensated deadline is %COMPENSATED_DEADLINE%</b><br/>
%endif%
%if:EXPLANATION_CAN_BE_SENT = 'true'%
<b>You must provide an explanation for this late deliverable in OR!</b><br/>
Deadline for providing an explanation: %EXPLANATION_DEADLINE%
%endif%
</p>
```

4.3.5 Sample not responded late deliverable PM notification email body template

pm_notification_email_template.html

```
<p>
You have explained but not responded late deliverable record(s) for your projects. See
details below.
</p>
%loop:RECORDS%
<p>
Late deliverable ID: %LATE_DELIVERABLE_ID%<br/>
Late deliverable type: %LATE_DELIVERABLE_TYPE%<br/>
Deliverable: %DELIVERABLE_NAME% (%PHASE_NAME% phase)<br/>
Deadline: %DEADLINE%<br/>
%if:COMPENSATED_AND_REAL_DEADLINES_DIFFER = 'true'%
Compensated deadline: %COMPENSATED_DEADLINE%<br/>
%endif%
Delay: %DELAY%<br/>
Contest link: <a
href="http://software.topcoder.com/review/actions/ViewProjectDetails.do?pid=%PROJECT_ID%">%PR
OJECT_NAME% %PROJECT_VERSION%</a>
```

```
</p>
%endloop%
```

4.3.6 Usage scenario

In this section it's assumed that configuration provided in previous sections is used.

This section describes how "Missed Deadline" late deliverable is tracked and notified. Tracking and notification for "Rejected Final Fix" late deliverables is performed similarly.

Assume that Review phase of Sample Project with ID=1 is currently late because one reviewer didn't submit a review scorecard in time (and deadline was not compensated). Then after executing LateDeliverablesTracker (via command line utility or programmatically), the following record can be added to the database:

late_deliverable table
(split, not important columns are skipped)

late_deliverable_id	project_phase_id	resource_id	deliverable_id	deadline
1	102	34	4	2010-08-26 09:05:00

late_deliverable_id	create_date	forgive_ind	last_notified	delay
1	2010-08-26 09:07:26	0	2010-08-26 09:07:26	146

late_deliverable_id	compensated_deadline	explanation	response	late_deliverable_type_id
1	NULL	NULL	NULL	1

Additionally the following email message must be sent:

```
--------------------------------------------------------------------------------
From: service@topcoder.com
To: user1@topcoder.com
Subject: WARNING: You are late when providing a deliverable for Sample Project
--------------------------------------------------------------------------------
<p>
Deadline: 2010-08-26 09:05:00<br/>
Your current delay is: 2 minutes<br/>
Contest link: <a href="http://software.topcoder.com/review/actions/ViewProjectDetails.do?pid=1"
>Sample Project</a><br/>
<b>You must provide an explanation for this late deliverable in OR!</b><br/>
Deadline for providing an explanation: 2010-08-27 09:07:30
</p>
--------------------------------------------------------------------------------
```

If LateDeliverablesTracker will be executed in 5 minutes, it should update "delay" field of late_deliverable table with late_deliverable_id = 1. The new "delay" field value must be 446 (seconds). Since the notification interval value specified in 4.3.3 (600 seconds) is not yet reached, another notification email message should not be sent.

4.3.7 Compensated deadline demo

In this section it's assumed that length of Appeals phase is 24 hours and length of Appeals Response phase is 12 hours.

Assume that the Appeals phase started at 09:00 on 2010-12-01, so originally the Appeals Response phase was scheduled to start at 09:00 on 2010-12-02. But since all submitters used "Complete Appeals" OR feature the Appeals phase ended earlier at 16:32 on 2010-12-01 and the Appeals Response phase started immediately.

In this case the compensated deadline for the Appeals Response phase will be 2010-12-02 21:00 (it's scheduled end time before the Appeals phase ended). So if one of reviewers doesn't commit review scorecard in time (i.e. before the compensated deadline) he can get the following notification email message (note that delay is calculated taking the real deadline into account, not the compensated one):

```
--------------------------------------------------------------------------------
From: service@topcoder.com
```

```
To: reviewer1@topcoder.com
Subject: WARNING: You are late when providing a deliverable for Sample Project
--------------------------------------------------------------------------------
<p>
Deadline: 2010-12-02 04:32:15<br/>
Your current delay is: 16 hours 29 minutes<br/>
Contest link: <a href="http://software.topcoder.com/review/actions/ViewProjectDetails.do?pid=1"
>Sample Project</a><br/>
<b>Note that the deadline was compensated due to the dependency phases having ended
prematurely.<br/>
The compensated deadline is 2010-12-02 21:00:00</b></br>
<b>You must provide an explanation for this late deliverable in OR!</b>
Deadline for providing an explanation: 2010-12-03 21:01:32
</p>
--------------------------------------------------------------------------------
```

At the same time please note that the Review phase time won't be compensated when the Screening phase ends earlier since the Review phase (in design and development contests) is usually 48 hours long.

4.3.8 *PM notification about not responded late deliverables demo*

Assume that DB contains the following information about late deliverables:

late_deliverable table
(split, not important columns are skipped)

late_deliverable_id	deliverable_id	deadline	compensated_deadline
1	4	2011-02-26 09:05:00	NULL
2	11	2011-03-01 12:01:00	2011-03-01 17:05:00
3	20	2011-03-08 13:00:00	NULL
4	22	2011-03-09 18:42:00	2011-03-09 19:05:00
5	20	2011-03-09 20:15:00	NULL

late_deliverable_id	delay	explanation	response	late_deliverable_type_id
1	146	Dog ate my laptop.	NULL	1
2	1285	NULL	NULL	1
3	5904	Waiting for docs from the client.	Forgiven.	1
4	4580	Waiting for response from PM.	NULL	1
5	NULL	New requirement is added	NULL	2

Assuming that all these late deliverables correspond to the same manager, and NotRespondedLateDeliverablesNotifier#execute() is called, the following email message is expected to be sent:

```
--------------------------------------------------------------------------------
From: service@topcoder.com
To: manager@topcoder.com
Subject: WARNING: You have explained late deliverable(s) to be responded
--------------------------------------------------------------------------------
<p>
You have explained but not responded late deliverable record(s) for your projects. See
details below.
</p>
<p>
Late deliverable ID: 1<br/>
Late deliverable type: Missed Deadline<br/>
Deliverable: Review Scorecard (Review phase)<br/>
Deadline: 2011-02-26 09:05:00<br/>
Delay: 2 minutes<br/>
Contest link: <a
href="http://software.topcoder.com/review/actions/ViewProjectDetails.do?pid=1">Sample Project
1.0</a>
```

```
</p>
<p>
Late deliverable ID: 4<br/>
Late deliverable type: Missed Deadline<br/>
Deliverable: Final Review (Final Review phase)<br/>
Deadline: 2011-03-09 18:42:00<br/>
Compensated deadline: 2011-03-09 19:05:00<br/>
Delay: 1 hour 16 minutes<br/>
Contest link: <a
href="http://software.topcoder.com/review/actions/ViewProjectDetails.do?pid=1">Sample Project
1.0</a>
</p>
<p>
Late deliverable ID: 5<br/>
Late deliverable type: Rejected Final Fix<br/>
Deliverable: Final Fix (Final Fix phase)<br/>
Deadline: N/A<br/>
Compensated deadline: N/A<br/>
Delay: N/A<br/>
Contest link: <a
href="http://software.topcoder.com/review/actions/ViewProjectDetails.do?pid=1">Another
Project 1.1</a>
</p>
--------------------------------------------------------------------------------
```

4.3.9 Sample notification email body template for Rejected Final Fix late deliverable

rejected ff email template.html

```
<p>
Contest link: <a
href="http://software.topcoder.com/review/actions/ViewProjectDetails.do?pid=%PROJECT_ID%">%PR
OJECT_NAME%</a><br/>
<b>You must provide an explanation for the rejected Final Fix in OR!</b><br/>
</p>
```

5. Future Enhancements

None