

This page last changed on Apr 03, 2008 by [ghostar](#).

## 1. Scope

### 1.1 Overview

The TopCoder Client Cockpit application defines a set of phase types. This component provides the plug-ins to the Phase Management component, whose logic is to check if these phases can be executed. Extra logic to execute the phases is also provided. This component is similar to the Online Review Phases component.

#### 1.1.1 Version

1.0

### 1.2 Logic Requirements

#### 1.2.1 Contest Information

The phase handlers will utilize the ContestManager interface, from the Studio Contest Manager component, which will be deployed in an EJB 3.0 container. The constructors for the phase handlers should expect either a ContestManagerRemote bean being directly provided, or a name that can be used with JNDI to retrieve the bean.

#### 1.2.2 Phase Handler interface

Each phase handler class must implement the PhaseHandler interface from the Phase Management component.

#### 1.2.3 Phase Change Emails

Upon any phase change, emails should be sent to the resources associated with timeline notification for the project. The email template should be configurable.

#### 1.2.4 Changing Phases

To change a contest's phase, use the ContestManager to get all contest statuses (getAllContestStatuses). Find the status that has a description taken from the table below and apply it to the contest, committing the change using ContestManager.updateContest.

Phase name	Contest status description
Draft	Draft
Scheduled	Scheduled
Active	Active
Action Required	ActionRequired
In Danger	InDanger
Insufficient Submissions - ReRun Possible	InsufficientSubmissionsReRunPossible
Extended	Extended

Repost	Repost	
Insufficient Submissions	InsufficientSubmissions	
No Winner Chosen	NoWinnerChosen	
Completed	Completed	
Abandoned	Abandoned	
Cancelled	Cancelled	
Terminated	Terminated	

## 1.2.5 Draft Phase Handler

This phase will start as soon as the contest is created.

The following cases will end the phase:

- When the client pays for the contest, the Scheduled phase is started

## 1.2.6 Scheduled Phase Handler

The Scheduled phase will start as soon as a contest with "Draft" status has been paid for.

The following cases will end the phase:

- When the Contest.startDate time has been reached, the Active phase is started

## 1.2.7 Active Phase Handler

The Active phase can start as soon as a contest with "Scheduled" status has its startDate property time reached.

The following cases will end the phase:

- When the Contest.endDate has been reached, and the minimum number of submissions has been reached, the Action Required phase is started
- When the Contest.endDate has been reached, the minimum number of submissions has not been reached, the Insufficient Submissions - ReRun Possible phase is started

## 1.2.8 Action Required Phase Handler

The "Action Required" phase will start as soon as a contest ends and the minimum number of submissions has been met.

The following cases will end the phase:

- When the client has chosen a winner, the Completed phase is started. This can be determined using the Contest.results property.
- When the client decides not to choose a winner, the No Winner Chosen phase is started
- When less than 24 hours remains before the Contest.winnerAnnouncementDeadline date, the In Danger phase is started.

## 1.2.9 In Danger Phase Handler

This phase starts when a contest has less than 24 hours before the client is supposed to pick a winner.

The following cases will end the phase:

- When the client has chosen a winner, the Completed phase is started. This can be determined using the Contest.results property.
- When the Contest.winnerAnnouncementDeadline has been reached without a winner chosen, the Abandoned phase is started.

The following actions are also necessary during the processing of this phase:

- When less than 8 hours remain before the Contest.winnerAnnouncementDeadline, a configurable reminder email should be sent to the client
- When less than 1 hour remains before the Contest.winnerAnnouncementDeadline, a different configurable reminder email should be sent to the client

## **1.2.10    *Insufficient Submissions - ReRun Possible Phase Handler***

This phase is started when a contest has ended, the minimum number of submissions has not been reached, and the contest is not a rerun

The following cases will end this phase:

- When the client chooses to rerun the contest, the "Extended" phase is started

## **1.2.11    *Extended Phase handler***

This phase is started if a contest fails to get the required number of submissions during its first run-through.

The following cases will end the phase:- When the Contest.endDate has been reached, and the minimum number of submissions has been reached, the Action Required phase is started

- When the Contest.endDate has been reached, the minimum number of submissions has not been reached, the Insufficient Submissions phase is started

## **1.2.12    *Repost Phase Handler***

This phase is reached when a contest fails to generate the required number of submissions both during an initial posting, as well as after a rerun. It can also be reached when a contest finishes and generates the required number of submissions, but the client decides not to choose a winner. At both of these cases, the client has the option of reposting the contest.

The following cases will end the phase

- When the Contest.endDate has been reached, and the minimum number of submissions has been reached, the Action Required phase is started
- When the Contest.endDate has been reached, the minimum number of submissions has not been reached, the Insufficient Submissions - ReRun Possible phase is started

## **1.2.13    *Insufficient Submissions Phase Handler***

This phase is reached if a contest completes with less than the minimum number of submissions, and the contest has already been rerun.

The following cases will end the phase

- When the client chooses to repost the contest, the Repost phase is started
- When the client abandons the contest, the Abandoned phase is started
- When the client cancels the contest, the Cancelled phase is started

## **1.2.14    *No Winner Chosen Phase Handler***

This phase is reached when a contest completes successfully, with the required number of submissions, but the client decides not to choose any submissions as a winner. The following cases will end the phase

- When the client chooses to cancel the contest, the Cancelled phase is started
- When the client chooses to repost the contest, the Repost phase is started

## **1.2.15    *Completed Phase Handler***

This phase is reached when a contest completes successfully, with the required number of submissions, and the client chooses a winner. This phase is an end phase.

## **1.2.16    *Abandoned Phase Handler***

This phase is reached when a contest completes successfully, with the required number of submissions, and the client takes no action. This phase is an end phase.

## **1.2.17    *Cancelled Phase Handler***

This phase is reached when a client chooses to cancel a contest. This phase is an end phase,

## **1.2.18 Terminated Phase Handler**

This phase is reached when TopCoder chooses to terminate a contest. This phase is an end phase.

## **1.3 Required Algorithms**

There should be a detailed discussion provided explaining how the end cases for each phase are determined.

## **1.4 Example of the Software Usage**

The Client Cockpit application will plug the phases into the Phase Management component, which will be used by the Auto Pilot component to automatically move contests to the correct phases.

## **1.5 Future Component Direction**

None.

# **2. Interface Requirements**

## **2.1.1 Graphical User Interface Requirements**

None.

## **2.1.2 External Interfaces**



## **2.1.3 Environment Requirements**

- Development language: Java1.5
- Compile target: Java1.5, Java1.6

## **2.1.4 Package Structure**

com.topcoder.clientcockpit.phases

# **3. Software Requirements**

## **3.1 Administration Requirements**

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

- The connection to the ContestManager bean

## 3.2 Technical Constraints

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

- JNDI

### 3.2.2 TopCoder Software Component Dependencies:

- Studio Contest Manager 1.0
- Contest and Submission Entities 1.0
- Phase Management 1.0
- Email Engine 3.0
- Document Generator 2.1

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

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

None.

### 3.2.4 QA Environment:

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

### 3.2.5 Database Connections

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

## 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 Poseidon.