# Time Tracker Report 2.0 Requirements Specification

## 1. Scope

### 1.1 Overview

The Time Tracker Report custom component is part of the Time Tracker application. It provides the ability to create time and expense reports for employees, projects, and clients. This component handles the necessary database queries that make up the various reports.

Version 2.0 will leverage the version 1.0 design by modifying existing and adding new classes and methods to meet the new requirements.

Note: the name of the package has changed. Existing table names have changed, but their meanings have not.

### 1.2 Logic Requirements

#### *1.2.1 Existing Requirements*

This version of Time Tracker Report will assume all previous requirements from version 1.0. Refer to the Expense Entry 2.0, Time Entry 2.0, Time Tracker Project 2.0 and Time Tracker User 2.0 components for the updated database schemas and the meanings of the columns.

#### *1.2.2 Company Account*

The new version of the Time Tracker application introduces a new entity called company.
A company owns everything in the context of the application, such as users, clients, and projects.

In generally, each report will support an additional filter for company IDs. It will be possible to specify multiple company IDs. The company ID's will be set automatically by the presentation layer. As an option, the company IDs will be generated as the first column of the report.

Note: some or all of the existing SQL queries will no longer be valid. The new version will need to fix them.

#### *1.2.3 Header and Footer*

Each report will have a header and footer section.

For worker reports, the header will display "[Username] [Start Date] to [End Date]".
For project reports, the header will display "[Project Name] [Start Date] to [End Date]".
For client reports, the header will display "[Client Name] [Start Date] to [End Date]".

For expense reports, the footer will display the total expenses under the expense column.
For time reports, the footer will display the total hours under the hours column.

The user will be able to specify whether to display the header or the footer,

#### *1.2.4 Column Sorting*

The design will allow the application or presentation layer to specify column sorting of the generated report. Sorting can be based on single column or multiple columns. Sorting order can be ascending or descending.

#### *1.2.5 Date Range Filters*

The version 1.0 implementation requires both the Start Date and End Date to be specified for the date range. In fact, the range may be open-ended. i.e. Either the Start Date or End Date can be left unspecified. The new version will provide the correct behavior.

#### *1.2.6 Empty Value Display*

For the Hours and Amount columns, it is possible that the table cell is empty or null. The version 1.0 implementation displays the “*” character. The new version will make the display for empty or null cells a configurable option.

### 1.3 Required Algorithms

None.

### 1.4 Example of the Software Usage

The Time Tracker application will use this component to generate various time and expense reports for employees, projects, and clients.

### 1.5 Future Component Direction

Other database systems maybe plugged in for some client environments. Additional reports may be implemented in the future.

## 2. Interface Requirements

#### *2.1.1 Graphical User Interface Requirement*

None.

#### *2.1.2 External Interfaces*

None.

#### *2.1.3 Environment Requirements*

- Development language: Java 1.4
- Compile target: Java 1.4, Java 1.5

#### *2.1.4 Package Structure*

com.cronos.timetracker.report

## 3. Software Requirements

### 3.1 Administration Requirements

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

- Display for empty or null table cells

## 3.2 Technical Constraints

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

None.

### *3.2.2 TopCoder Software Component Dependencies:*

- Configuration Manager
- DB Connection Factory

**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:*

Informix Database.

### *3.2.4 QA Environment:*

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

## 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.