Software Documentation : Java Custom COO Generator

This page last changed on Jan 05, 2010 by mashannon168.

1. Scope

1.1 Overview

In this contest, you will design a component that will act as a tool to generate a COO document. We have specific output requirements, but you are free to use varying techniques/components/open source/etc. to create the outputs. The basic idea is that the user will be able to go to enter or select a past TopCoder contest and the component will query TC data to dynamically include in a web page or PDF file.

1.1.1 Version

1.0

1.2 Logic Requirements

This component will remove the need to manually create COO PDF documents. For this contest, we will only deal with non-Studio contests.

1.2.1 Retrieve Contest Data

- You will need to come up with the queries needed to retrieve the data. You can copy/reference from these queries which are used by tc http://www.topcoder.com/tc?module=CompContestDetails&pj=

find reviewers

```
 select distinct sr.reviewer_id, sr.review_resp_id, rr.review_resp_desc
from submission_review sr
, review_resp rr
where sr.project_id = @pj@
and sr.review_resp_id is not null
and sr.review_resp_id = rr.review_resp_id  order by sr.review_resp_id
```

project info

```
select
component_name,
component_id,
version_id,
phase_id,
version_text,
category_desc,
phase_desc,
status_id,
num_registrations as num_inquiries,
num_submissions,
case num_registrations
WHEN null then null
WHEN 0 then null
else 100.0 * num_submissions  / num_registrations
end as submission_percent,
num_valid_submissions as passed,
case num_submissions
WHEN 0 then null
else 100.0 * num_valid_submissions / num_submissions
end as passed_percent,
avg_raw_score,
avg_final_score,
project_id, viewable_category_ind,
category_id
```

```
from project
where project_id = @pj@
```

submissions

```
select user_id,
inquire_timestamp,
submit_timestamp,
(select final_score from submission_screening ss where project_id = pr.project_id and user_id =
pr.user_id) as screening_score,
raw_score as initial_score,
final_score,
points_awarded as initial_points,
final_points,
valid_submission_ind as passed_screening,
(select final_score from submission_review sr where project_id = pr.project_id and user_id =
pr.user_id
and review_resp_id in (1,4)) as score1,
(select final_score from submission_review sr where project_id = pr.project_id and user_id =
pr.user_id
and review_resp_id in (2,5)) as score2,
(select final_score from submission_review sr where project_id = pr.project_id and user_id =
pr.user_id
and review_resp_id in (3,6)) as score3
from project_result pr
where project_id = @pj@
and submit_ind = 1
order by valid_submission_ind desc, final_score desc, screening_score desc
```

If you can find an existing component or EJB that provides the data then you should use it.

- You will need to map the 3rd party xml file into a table (provide scripts). Note that there are different worksheets for different catalogs.

Required data:

- Name of the contest with the version # appended
- End Date of the contest (ex. Sept. 4, 2009)
- All 1st, 2nd place and review board members must be listed here along with there role on the project. If the contest is a component development contest, then the design members as well as the development members must be listed.

Sample output:

Role	TopCoder Member
Design Winner	AleaActaEst
Design 2nd Place	Luca
Design Reviewer	Rica
Design Reviewer	Telly12
Design Reviewer	Linwe
Development Winner	Sparemax
Development 2nd place	EveningSun
Development Reviewer	bbxiong
Development Reviewer	Velorien

Development Reviewer	FireIce

- Lists of all 3rd party dependencies. The dependencies can be found in the build-dependencies.xml file in subversion in the trunk folder. The subversion path is a property on the Online Review project. (You can assume you can use a svn user that has access to all assets.) The dependencies should be used to lookup the information for the name, version, description, source URL, and License, from the third_party_libraries.xls file. See attched sample file.

If there are no third-party dependencies in the build file, the answer to the question below (Does the build file have any third party dependencies) should be "no." If there are third party dependencies, the answer should be "yes."]

If there are third party dependencies, but they are not found in the spreadsheet, the report should highlight the problem by listing the description, Source URL and License as "NOT FOUND".

Sample 3rd part table:

Name	Version	Description	Source URL	License
Antlr	2.7.6	Parser Generator	www.antlr.org	Antlr 3 License (BSD-style)
JUnit	3.8.2	Unit Testing Framework	www.junit.org	CPL

1.2.2 Generate XML Output

Provide a generic xml format to output the data. This can be used by other client programs to retrieve the data and manipulate it on their side.

1.2.1 Generate PDF Output

- The PDF output should be template based.
- The default name of the file should be "COO_CONTEST_ID.pdf", where CONTEST_ID is the project_id.
- Follow the layout of the attached modelCOO-for-automation.doc. Note the comments in blue which describe the data sections.
- You can use http://xmlgraphics.apache.org/fop/, let us know if you think there is a better one.
- We need two methods: one is to save to a pdf file, another is to generate the bytes/stream so that it can be used to download/save to a local file.

1.3 Required Algorithms

None.

1.4 Example of the Software Usage

Used by TC to genereate file when a contest is completed.

1.5 Future Component Direction

None.

2. Interface Requirements

2.1.1 Graphical User Interface Requirements

None.

2.1.2 External Interfaces

See 1.1

2.1.3 Environment Requirements

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

2.1.4 Package Structure

com.topcoder.management.contest.coo

3. Software Requirements

3.1 Administration Requirements

3.1.1 What elements of the application need to be configurable?

svn user/password.

Any template files (path).

Path to where pdf will be saved.

3.2 Technical Constraints

3.2.1 Are there particular frameworks or standards that are required?

3.2.2 TopCoder Software Component Dependencies:

**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:

Apache FOP. http://xmlgraphics.apache.org/fop/

3.2.4 QA Environment:

- Solaris 7
- RedHat Linux 7.1
- Windows XP
- 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.