

Java Swing Side Menu 1.0 Requirements Specification

1. Scope

1.1 Overview

The Java Swing Side Menu component provides a panel in a Swing application that contains a set of controls that can be opened or closed by a user. The menu takes up the full left or right side of an application. This component will be used to house functionality in the UML Tool that isn't always necessary. The user can choose to collapse the side menu to allow for more space for a diagram being viewed.

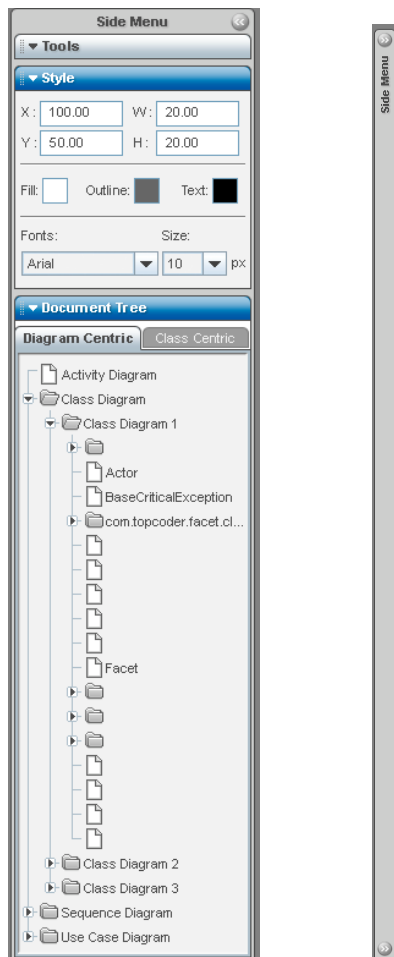
1.2 Logic Requirements

1.2.1 Control layout

Prototype images are given that explain the overall layout and functionality of the side menu. Note that the custom look and feel **is not** part of the competition, as that will be handled externally.

1.2.2 Side menu functionality

The side menu is a control that can be put at the left or right side of an application. It contains a single panel that can be manipulated to contain the inner controls for the menu. The side menu can be collapsed to take up very little space, or it can be expanded to shown the necessary panel



and controls the panel contains. No animation is needed for the expanding and collapsing, but if some is provided it would be considered additional functionality, as long as the animation can be turned off, if needed. The images show the side menu expanded.

The width of the expanded menu is programmatically set and can't be changed by the application user.

1.2.3 Events

An event handler interface should be defined to allow the user to perform actions when the side menu is collapsed or expanded.

1.2.4. Side menu layout

For the initial version of this component, the interior of the side menu will just be a panel that the user can set the layout manager for and add and remove controls from. The layout seen in the above images will be part of a separate component.

1.3 Required Algorithms

None

1.4 Example of the Software Usage

This component will be used in the TopCoder UML Tool to house functionality for modifying elements, but allowing the user to hide the panels to allow more space for the diagram view.

1.5 Future Component Direction

None

2. Interface Requirements

2.1.1 Graphical User Interface Requirements

The look of the side menu and text orientation should match the prototype given, using placeholders for images. The individual panel header styles and side menu style do not have to match the prototype, as the exact look and feel will be updated.

2.1.2 External Interfaces

None

2.1.3 Environment Requirements

- ☐ Development language: Java 1.5
- ☐ Compile target: Java 1.5

2.1.4 Namespace

com.topcoder.gui.sidemenu

3. Software Requirements

3.1 Administration Requirements

3.1.1 What elements of the application need to be configurable?

- ☐ The width of the expanded side menu

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 2.1.5

****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

3.3 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

XML documentation must provide sufficient information regarding component design and usage.