Memory Usage 2.0 Requirements Specification

1. Scope

1.1 Overview

The Memory Usage component is used to obtain runtime memory usage detail for target objects. The component accepts an object and determines the total memory used. If specified, the component also determines the memory usage of each embedded object. The Memory Usage component can be used as a quick analysis tool to obtain a rough estimate of memory usage without incurring the overhead of executing a complete profiling tool.

The Memory Usage 1.0 analyzers are hard coded to specific JVM versions. The 2.0 version will add a generic fallback analyzer, as well as new JVM specific analyzers.

1.2 Logic Requirements

1.2.1 Sun Analyzers

The Sun121314Analyzer must be updated to include Sun's 1.5 JVM.

1.2.2 IBM Analyzers

Analyzers for IBM's 1.4 and 1.5 JVMs must be created.

1.2.3 Generic Analyzer

For any JVM that does not match the components specialized analyzers, the component must have a fallback generic analyzer version. This generic version should make an approximation based on Sun 1.4 behavior.

The component must allow the consumer to decide whether or not to use fallback analysis, and which analyzer to use. The default behavior must be to use fallback and the generic analyzer, to allow the component to work when deployed to new JVMs.

1.2.4 Update to Current Standards

The component is an older component and the 2.0 version must be updated to current TCS standards in every regard (diagrams, interface standards, etc.).

1.3 Required Algorithms

The designer must specify object size approximation algorithms for all new analyzers.

1.4 Example of the Software Usage

A cache may evict items based on the overall size of the cache. As items are added, the size must be determined.

1.5 Future Component Direction

None.

2. Interface Requirements

2.1.1 Graphical User Interface Requirements

None.

2.1.2 External Interfaces

None.

2.1.3 Environment Requirements

- Development language: Java1.5
- Compile target: Java1.4

2.1.4 Package Structure

com.topcoder.util.memoryusage

3. Software Requirements

3.1 Administration Requirements

3.1.1 What elements of the application need to be configurable?

None.

3.2 Technical Constraints

3.2.1 Are there particular frameworks or standards that are required?

The analyzers should adhere to the JVM defined memory approximation methods.

3.2.2 TopCoder Software Component Dependencies:

None.

**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:

IBM JVM 1.4 and 1.5.

3.2.4 QA Environment:

- 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.

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.