[TOPCODER]
SOFTWARE

Requirements Specification

1. Scope

1.1 Overview

A Universally Unique Identifier (UUID) or GUID (Globally Unique Identifier) is a unique identifier can be generated without a central authority and is unique across all servers. Since a central authority is not needed, the identifiers will be generated without using any persistent data store such as a file or database. A 128 bit UUID is guaranteed to be unique for over a thousand years.

1.2 Logic Requirements

1.2.1 Generate a Universally Unique Identifier (UUID) in memory

- The algorithm will not require a persistent data store such as a database or file.

1.2.2 UUID length

- An official UUID is a 128 bits long
  - Provide a function to generate a 128 bit UUID.
- A 32 bit UUID is not as guaranteed to be unique as long as a 128 bit UUID, but is still useful in many circumstances.
  - Provide a function to generate a 32 bit UUID.

1.2.3 Rate of Generation

The component will be capable of generating thousands of GUIDS a second. This will allow the component to be used in a transactional system if necessary.

1.2.4 Example algorithm

- This is the working standard for UUID generation.
  - http://www1.ics.uci.edu/~ejw/authoring/uuid-guid/draft-leach-uuids-guids-01.txt
- This website displays an example algorithm.
  - http://www.jguru.com/faq/view.jsp?EID=1030397

1.3 Required Algorithms

Supply an algorithm to generate a UUID without using a database or file.

1.4 Example of the Software Usage

A session identifier needs to be guaranteed unique. This identifier needs to be created quickly and efficiently. This identifier would uniquely identify a user's web session.

1.5 Future Component Direction

None.

2. Interface Requirements

2.1.1 Graphical User Interface Requirements

None.

2.1.2 External Interfaces

Generate a new GUID.

2.1.3 Environment Requirements

- Development language: Java1.4
- Compile target: Java1.3, Java1.4

2.1.4 Package Structure

com.topcoder.util.generator.guid

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?

Java 1.3 and 1.4.

3.2.2 TopCoder Software Component Dependencies:

IDGenerator
**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

- Design documents must clearly define intended component usage in the 'Documentation' tab of Poseidon.