GMU-Team 1 : Documentation Report

IDT Spring 2014

Brannon Crymes - Akshay Karthik – Sriram Rajaraman

2014

Contents

[Introduction 2](#_Toc377290315)

[Running the System 2](#_Toc377290316)

[Usage 3](#_Toc377290317)

[Testing API 3](#_Toc377290318)

[Log Viewer 3](#_Toc377290319)

[GMUT Tests 3](#_Toc377290320)

[Architecture 4](#_Toc377290321)

[GMUT 4](#_Toc377290322)

[Test Builder 4](#_Toc377290323)

[Predicates 4](#_Toc377290324)

[Report Writer 4](#_Toc377290325)

[Log Viewer 4](#_Toc377290326)

[Requirements Traceability 5](#_Toc377290327)

# Introduction

George Mason University - Team 1  
Brannon Crymes - [bwinter2@gmu.edu](mailto:bwinter2@gmu.edu)  
Akshay Karthik - [akshay.karthik@gmail.com](mailto:akshay.karthik@gmail.com)  
Sriram Rajaraman - [s.rajaaraman89@gmail.com](mailto:s.rajaaraman89@gmail.com)

We were asked to implement a system that allows developers to place checks within their code that can record the results of their code and asses the logic of the system. We call our system GMUT or GMU Tester. When we architected the system, we focused on simplicity of use, extensibility, and performance. We used plain java with no additional libraries to minimize the footprint of our system. The system is well documented and tested and fails gracefully if the application crashes. It does not leak memory and its thread safe API makes it easy to use in a variety of applications.

## Running the System

The logging framework can be run directly through the com.idt.contest.college.winter2014.Main interface. It works in both batch script mode as well as menu driven mode. The system returns results in a log.txt file that follows a specific format. This format was chosen to be able to search through via regular expressions and is documented in documentation/report\_format.txt. This format is used in logviewer as well as all of the included ReportWriters however, the extensible nature of the system allows for custom ReportWriters in any format that a developer wishes to use. There are three key access points to the system:

1. gmu\_main.jar is the com.idt.contest.college.winter2014 jar with the functions annotated by the GMUT testing API. It can be run without arguments to access the menu driven application. If an argument is provided, it runs using a batch script.   
   **java –jar gmu\_main.jar  
   java –jar gmu\_main.jar batchscript.txt**
2. gmu\_logviewer.jar is the application to view the logs.

**java –jar gmu\_logviewer.jar**

1. gmu\_tests.jar is the unit tests for the GMUT API. It is a simple console application that runs all of the unit tests that verify that the system meets the given requirements.

**java –jar gmu\_tests.java**

## Building the System

Building the system is simple, it can be built from eclipse.

1. Import project into eclipse (file > import existing project)
2. Run Configurations
   1. LogViewer.java for log viewer
   2. Main.java for the IDT application
   3. InternalTestRunner.java for the internal tests application

Also provided is an Ant build.xml file for Apache Ant build system. This system automatically builds all three jars as well as places the default batchfile.txt file in the main directory.

To run the ant build file, simply run the following console command. This assumes that java as well and ant is available on the console’s PATH.

**ant -buildfile test.xml**

The ant file performs four actions:

1. Build /src into gmu\_main.jar with com.idt.contest.college.winter2014.Main as the main class.
2. Build /src into gmu\_logviewer.jar with edu.gmu.team1.idt2014.logviewer.LogViewer as the main class
3. Build /test into gmu\_tests.jar with edu.gmu.team1.idt2014.test.InternalTestRunner as the main class
4. Copy batchscript.txt into the main directory.

# Usage

## Testing API

## Log Viewer

## GMUT Tests

# Architecture

## GMUT

## Test Builder

## Predicates

## Report Writer

## Log Viewer

# Requirements Traceability