

# RUnit - A Unit Test Framework for R

Thomas König, Klaus Jünemann, and Matthias Burger

May 17, 2004

## Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	Motivation . . . . .	2
1.2	Background . . . . .	2
<b>2</b>	<b>Getting Started: Setting up test cases</b>	<b>2</b>
2.1	R Unit Testing . . . . .	2
2.2	R Code Inspection . . . . .	2
<b>3</b>	<b>Summary</b>	<b>2</b>
<b>4</b>	<b>Future Development Ideas</b>	<b>2</b>

## Abstract

Software development for production systems presents a challenge to the development team as the quality of the coded package(s) has to be constantly monitored and verified. We present a generic approach to software testing for the R language modelled after successful examples such as JUnit, CppUnit, and PerlUnit. The aim of our approach is to facilitate development of reliable software packages and provide a set of tools to analyse and report the software quality status. The presented framework is completely implemented with R and does not rely on external tools or other language systems. The basic principle is that every function or method is accompanied with a test case that queries many calling situations including incorrect invocations. A test case can be executed instantly without reinstalling the whole package - a feature that is necessary for parallel development of functionality and test cases. On a second level one or more packages can be tested in a single test run, the result of which is reported in an easy to understand test protocol.

To verify the coverage of the test framework a code inspector is provided that monitors the code coverage of executed test cases. The result of individual test invocations as well as package wide evaluations can be compiled into a summary report exported to HTML. This report details the executed tests, their failure or success, as well as the code coverage. Taking it one step further and combining the build system with a development and release procedure with defined code status description this approach opens the way for a principled software quality monitoring and risk assessment of the developed application. For our code development we have utilized the described system with great benefit w.r.t. code reliability and maintenance efforts in a medium sized development team.

## **1 Introduction**

The *RUnit* package provides ...

### **1.1 Motivation**

### **1.2 Background**

## **2 Getting Started: Setting up test cases**

### **2.1 R Unit Testing**

### **2.2 R Code Inspection**

```
R> library(RUnit) ##load the package
```

## **3 Summary**

## **4 Future Development Ideas**