Table of Contents

[1 Introduction 3](#_Toc500097876)

[2 Public API Overview 3](#_Toc500097877)

[2.1 Public Classes 3](#_Toc500097878)

[2.2 Public Routines 3](#_Toc500097879)

[3 Test Suite Description 4](#_Toc500097880)

[3.1 Individual Tests 4](#_Toc500097881)

[3.1.1 General 4](#_Toc500097882)

[3.1.2 Presentation Semantics 5](#_Toc500097883)

[3.1.3 Document Structure 8](#_Toc500097884)

[3.1.4 Output 9](#_Toc500097885)

[4 Expected Coverage 10](#_Toc500097886)

[5 Benchmarks 10](#_Toc500097887)

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author(s)** |
| 2017-11-21 | 0.1 | Test cases and Test Suite description added | Elfat Esati |
| 2017-11-30 | 0.2 | Test cases, Public Classes and Routines Added |  |
| 2017-12-02 | 0.3 | Introduction,Set Up, Tear Down added | Elfat Esati |

# 

# Introduction

This Test Plan depicts the framework of the Markdup Library on how the library will be evaluated and under what circumstances it shall be released. We ought to ensure that our requirements are meeting with the expected product (library). To do that we will be showing on what is to be tested and how it should be tested. The Test Suite details the data and steps required to execute a single test case.

# Public API Overview

The «Markup Generator» is an Eiffel library for generating reports written in markup lan- gauges with the focus in html and markdown. It contains several subsystems to satisfy all requirements. But there are not only interfaces between the subcomponents but also between the library and the user. This user interface will be realized in the form of an API that allows other Eiffel programs to take advantage of the library’s functionalities.

## Public Classes

The following classes are part of the public API and should be tested by using automated test procedures:

|  |  |
| --- | --- |
| **Container Elements** | **Leaf Elements** |
| * DOCUMENT * LISTNODE * TABLENODE * TEBLEROWNODE | * CUSTOMELEMENT * TEXTNODE * DOCUMENTLINKNODE * LISTITEMNODE * TABLECELLNODE * IMAGENODE * HEADIGNODE * EXTERNALLINKNODE |

## Public Routines

|  |  |  |
| --- | --- | --- |
| **Routine** | **Class of the Routine** | **Expected Arguments** |
| accept | DOCUMENT | Object belonging to a subclass of A\_VISTOR |
| getname | DOCUMENT | none |
| accept | LISTNODE | Object belonging to a subclass of A\_VISTOR |
| addChild | LISTNODE | Object of LISTITEMNODE |
| accept | TABLEROWNODE | Object belonging to a subclass of A\_VISTOR |
| accept | TABLENODE | Object belonging to a subclass of A\_VISTOR |
| addChild | TABLENODE | Object of TABLEROWNODE |
| accept | CUSTOMELEMENT | Object belonging to a subclass of A\_VISTOR |
| addtext | CUSTOMELEMENT | String |
| accept | TEXTNODE | Object belonging to a subclass of A\_VISTOR |
| accept | DOCUMENTLINKNODE | Object belonging to a subclass of A\_VISTOR |
| accept | LISTITEMNODE | Object belonging to a subclass of A\_VISTOR |
| accept | TABLECELLNODE | Object belonging to a subclass of A\_VISTOR |
| accept | IMAGENODE | Object belonging to a subclass of A\_VISTOR |
| getSource | IMAGENODE | none |
| getAlt | IMAGENODE | none |
| accept | HEADINGNODE | Object belonging to a subclass of A\_VISTOR |
| getLevel | HEADINGNODE | none |
| accept | EXTRENALLINKNODE | Object belonging to a subclass of A\_VISTOR |
| getLink | EXTRENALLINKNODE | none |

# Test Suite Description

Test Suite also called Validation Suite comprise the sets of test cases. Test Cases help us define or list down all the cases or situations under which the system or library works as expected and comply to a high level of quality. When the test cases represent a scenario for instance login, do something and log out, then the test suite may contain those test cases. Succesful completion of one test case need to happen before we test the next case. When we have identified all the test cases then we need to validate them to ensure the system works correctly with no failures and bugs.

## Individual Tests

### General

|  |  |
| --- | --- |
| Test Case ID | T.3.1.1.001 |
| SRS ID of Functionality | R3.2.1.001 |
| Name of Functionality | Inexplicit Usage of Functions |
| Routines | Public Routines |
| Set Up | Neccessary arguments to be passed to functions if needed. |
| Tear Down | None |
| Test Data |  |
| Oracle | The functions should have been implemented and as a result they must return what is expected from it. |

|  |  |
| --- | --- |
| Test Case ID | T.3.1.1.002 |
| SRS ID of Functionality | R3.2.1.002 |
| Name of Functionality | System |
| Routines |  |
| Set Up | None |
| Tear Down | None |
| Test Data |  |
| Oracle | If tested in different Eiffel IDE there is no guarantee to function properly. |

|  |  |
| --- | --- |
| Test Case ID | T.3.1.1.003 |
| SRS ID of Functionality | R3.2.1.003 |
| Name of Functionality | Creating Documents |
| Class and Routines | Class: Document, routine: make |
| Set Up | An instance of Document class should be declared. |
| Tear Down |  |
| Test Data |  |
| Oracle | A document object should be created. It should have an internal name that was determined within the creation procedure. No messages should be displayed. Furthermore, no errors should be raised. There is no item or value returned. |

### Presentation Semantics

|  |  |
| --- | --- |
| Test Case ID | T.3.1.2.001 |
| SRS ID of Functionality | R3.2.2.001 |
| Name of Functionality | Internal Links |
| Routines |  |
| Set Up |  |
| Tear Down |  |
| Test Data |  |
| Oracle | NOT YET IMPLEMENTED |

|  |  |
| --- | --- |
| Test Case ID | T.3.1.2.002 |
| SRS ID of Functionality | R3.2.2.002 |
| Name of Functionality | Linking Multiple Reports |
| Classes and Routines | Class: DOCUMENTLINKNODE, routine: make; Class: DOCUMENT, routine: addChild |
| Set Up | Two arguments to be passed to creation procedure of the class DOCUMENTLINKNODE. |
| Tear Down | None |
| Test Data | Document Link |
| Oracle | An object characterized by a link text and a DOCUMENT object, on which the object referrences, should be created and assigned to an object of type DOCUMENT. No messages should be displayed. Furthermore, no errors should be raised. There is no item or value returned. |

|  |  |
| --- | --- |
| Test Case ID | T.3.1.2.003 |
| SRS ID of Functionality | R3.2.2.003 |
| Name of Functionality | Linking Documents / Web Pages Accessible through the Web |
| Classes and Routines | Class: EXTERNALLINKNODE, routine: make; Class: DOCUMENT, routine: addChild |
| Set Up | The link (url address) to be provided |
| Tear Down | None |
| Test Data | Html link with ist content. |
| Oracle | An object characterized by a link text and a string representing an URL, on which the object refrences, should be created and assigned to an object of type DOCUMENT. No messages should be displayed. Furthermore, no errors should be raised. There is no item or value returned. |

|  |  |
| --- | --- |
| Test Case ID | T.3.1.2.004 |
| SRS ID of Functionality | R3.2.2.004 |
| Name of Functionality | Referencing Images by URL |
| Classes and Routines | Class: IMAGENODE and HTMLVISITOR, routines: make and makeAlt |
| Set Up | Image file with url link pointing to it. |
| Tear Down | None |
| Test Data | Image file |
| Oracle | An object characterized by an alternative text and a string representing an URL, which leads to the extrenal image, should be created and assigned to an object of type DOCUMENT. No messages should be displayed. Furthermore, no errors should be raised. There is no item or value returned. |

|  |  |
| --- | --- |
| Test Case ID | T.3.1.2.005 |
| SRS ID of Functionality | R3.2.2.005 |
| Name of Functionality | Generating Images from Local Files |
| Classes and Routines | Class: IMAGENODE and HTMLVISITOR, routines: make and makeAlt |
| Set Up | Image file address (with location where the image is). |
| Tear Down | None |
| Test Data | Image file |
| Oracle | An object characterized by an alternative text and a string representing a local file path, which leads to the local image, should created and assigned to an object of type DOCUMENT. No messages should be displayed. Furthermore, no errors should be raised. There is no item or value returned. |

|  |  |
| --- | --- |
| Test Case ID | T.3.1.2.006 |
| SRS ID of Functionality | R3.2.2.006 |
| Name of Functionality | Creating Bullet Lists |
| Classes and Routines | Class: LISTITEMNODE, routine: make; Class: LISTNODE, procedure: make, addChild; Class: DOCMENT, procedure: addChild |
| Set Up | Text as argument for the class LISTITEMNODE. |
| Tear Down | None |
| Test Data | Text |
| Oracle | One or multiple objects of type LISTITEMNODE containing simple text should be created and added to an object of type LISTNODE. The latter is assigned to an object of type DOCUMENT. The text contained in the objects of LISTITEMNODE are given by user input. No messages should be displayed. Furthermore, no errors should be raised. There is no item or value returned. |

|  |  |
| --- | --- |
| Test Case ID | T.3.1.2.007 |
| SRS ID of Functionality | R3.2.2.007 |
| Name of Functionality | Creating Tables |
| Routines | Class: TABLENODE;TABLEROWNODE,TABLECELLNODE, Routines: addChild, make. |
| Set Up | body as argument for creation procedure for class TABLECELLNODE. |
| Tear Down | None |
| Test Data | Text passed as argument. |
| Oracle | Table with columns and rows, filled by calling elements of the list in the TABLENODE. |

|  |  |
| --- | --- |
| Test Case ID | T.3.1.2.008 |
| SRS ID of Functionality | R3.2.2.008 |
| Name of Functionality | Text |
| Classes and Routines | Class: TEXTNODE, routine: make; Class: DOCUMENT, routine: addChild |
| Set Up | Text as argument for make of the class TEXTNODE. |
| Tear Down | None |
| Test Data | Text |
| Oracle | An object specified by a string entered by the user should be contained in an object of type DOCUMENT. No messages should be displayed. Furthermore, no errors should be raised. There is no item or value returned. |

|  |  |
| --- | --- |
| Test Case ID | T.3.1.2.009 |
| SRS ID of Functionality | R3.2.2.009 |
| Name of Functionality | Headers |
| Classes and Routines | Class: HEADINGNODE, routine: make; Class DOCUMENT, routine: addChild |
| Set Up | Text and Level argument for make of HEADINGNODE. |
| Tear Down | None |
| Test Data | Text provided |
| Oracle | An object of type HEADINGNODE which is characterized by two features representing the heading text and hierarchy level should be contained in an object of type DOCUMENT. No messages should be displayed. Furthermore, no errors should be raised. There is no item or value returned. |

|  |  |
| --- | --- |
| Test Case ID | T.3.1.2.010 |
| SRS ID of Functionality | R3.2.2.010 |
| Name of Functionality | Including Existing Code |
| Routines | Class: CUSTOMELEMENT, routines: make, addText |
| Set Up | Text as argument for make of the affected class. |
| Tear Down | None |
| Test Data | Given Text |
| Oracle | An object of type CUSTOMELEMENT containing an user-specified string representing markup code should be assigned to an oject of type DOCUMENT. No messages should be displayed. Furthermore, no errors should be raised. There is no item or value returned. |

|  |  |
| --- | --- |
| Test Case ID | T.3.1.2.011 |
| SRS ID of Functionality | R3.2.2.011 |
| Name of Functionality | Reading from File |
| Classes and Routines | Class: FILEWRITER |
| Set Up | Path to the file. |
| Tear Down | Delete File |
| Test Data | File read |
| Oracle | NOT YET IMPLEMENTED |

### Document Structure

|  |  |
| --- | --- |
| Test Case ID | T.3.1.3.001 |
| SRS ID of Functionality | R3.2.3.001 |
| Name of Functionality | Document Name |
| Classes and Routines | Class: DOCUMENT, routines: make, getName |
| Set Up | Document Name provided by user |
| Tear Down | None |
| Test Data |  |
| Oracle | Each object of type DOCMENT should be characterized by a user-specific string representing the document’s name. No messages should be displayed. Furthermore, no errors should be raised. There is no item or value returned. |

|  |  |
| --- | --- |
| Test Case ID | T.1. |
| SRS ID of Functionality | R3.2.3.002 |
| Name of Functionality | Correctly Indented Output |
| Classes and Routines | Class: HTML\_VISITOR, routine: getIndent, indent |
| Set Up | Class Instance of HTML\_VISITOR |
| Tear Down | None |
| Test Data | Document file |
| Oracle | The markup output should be well-structured by using indentation. Each child element should be intended in relation to its parent element. The indentation should always have the same size. No messages should be displayed. Furthermore, no errors should be raised. There is no item or value returned. |

### Output

|  |  |
| --- | --- |
| Test Case ID | T.3.1.4.001 |
| SRS ID of Functionality | R3.2.4.001 |
| Name of Functionality | Output to HTML (Clear Text) |
| Classes and Routines | Class: DOCUMENT, routine: accept |
| Set Up | List of children items only of html type. |
| Tear Down | Release of list from items. |
| Test Data | None |
| Oracle | The content an DOCUMENT object is returned in form of a string representing a valid HTML document. No messages should be displayed. Furthermore, no errors should be raised. |

|  |  |
| --- | --- |
| Test Case ID | T.3.1.4.002 |
| SRS ID of Functionality | R3.2.4.002 |
| Name of Functionality | Output to Markdown (Clear Text) |
| Routines | Class:MARKDOWN\_VISITOR, routine: visitDocument. |
| Set Up | Class instance for MARKDOWN\_VISITOR |
| Tear Down | Release if any |
| Test Data | None |
| Oracle | The content an DOCUMENT object should be returned in form of a string representing a valid markdown document. No messages should be displayed. Furthermore, no errors should be raised. |

|  |  |
| --- | --- |
| Test Case ID | T.3.1.4.003 |
| SRS ID of Functionality | R3.2.4.003 |
| Name of Functionality | Markup Output to Local File |
| Routines | Class: FILEWRITER, routine: write |
| Set Up | PLAIN TEXT FILE |
| Tear Down | None |
| Test Data | Text file |
| Oracle | A new text file should have been created, with the given path and content for it. |

# Expected Coverage

Testing is a constantly growing process, the provided Test Suite represents the test cases near the total coverage.

[What is the coverage you expect given your system and the individual tests you described?]

# Benchmarks

The benchmark testing aims to determine the execution time of each functional requirement that found its way into the «Markup Generator». This allows to perform statistical evaluations like the calculation of the mean execution time. Each benchmark test should be performed under the sam environmental conditions and there should not be any application in active state other than the ones needed for testing purposes. Furthermore, the hardware- and software specifications of the test system should be similar to the specifications of the user systems.