



# **RRT (Regression Report Tool)**

## *Application Note*

Revision 1.0

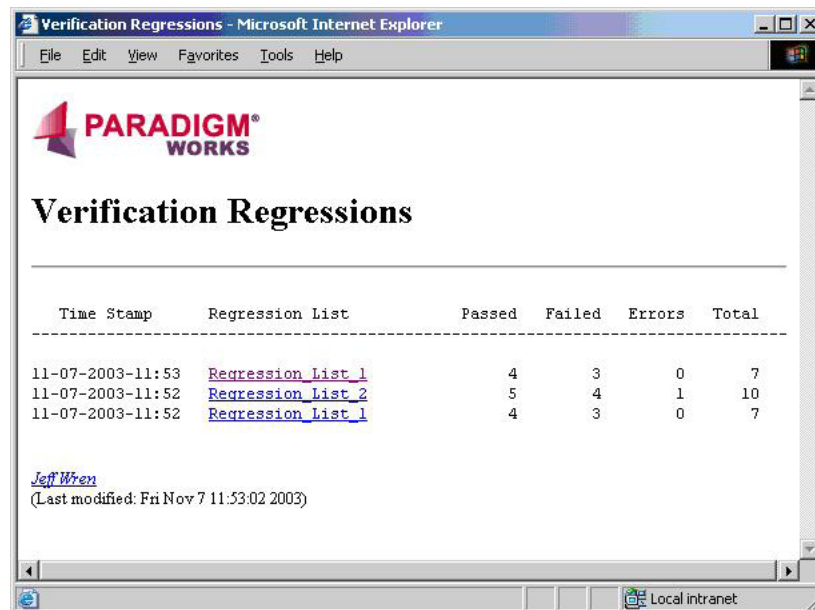
---

## Table of Contents

1	Overview .....	3
2	Features .....	3
3	Requirements.....	4
4	Configuration.....	4
4.1	Email Notification.....	4
4.2	Environment Configuration Parameters.....	4
4.3	Performance Configuration Parameters.....	5
4.4	Search Strings and Web Report Fields.....	5
5	Command Line Options .....	5
5.1	Address Override .....	5
5.2	Archiving .....	6
5.3	Configuration Files .....	6
5.4	Date Stamps .....	6
5.5	Deleting Old Archives.....	7
5.6	Deleting Old Reports.....	7
5.7	Email.....	7
5.8	Force.....	7
5.9	List Name .....	7
5.10	No Web Page.....	7
5.11	Parse Only .....	7
5.12	Recursive .....	7
5.13	Source Control Labels.....	7
6	Company Background .....	8

## 1 Overview

RRT (Regression Report Tool) is a Perl based tool that displays regression results via a web page. It takes as input a directory/list of test log files. It parses each for pass/fail information and any other information the user would like to have reported. It generates an HTML report that contains links to each of the tests allowing for quick navigation. (See **Figure 4-1: Sample Regression Report Page**). Finally, it automatically updates a master/home page where links to each of the reports resides (See **Figure 1-1** below).



Time Stamp	Regression List	Passed	Failed	Errors	Total
11-07-2003-11:53	<a href="#">Regression_List_1</a>	4	3	0	7
11-07-2003-11:52	<a href="#">Regression_List_2</a>	5	4	1	10
11-07-2003-11:52	<a href="#">Regression_List_1</a>	4	3	0	7

[Jeff Wren](#)  
(Last modified: Fri Nov 7 11:53:02 2003)

**Figure 1-1: Main Report Web Page**

## 2 Features

RRT includes the following features:

- Accepts as input a directory, a list of files, list of directories, etc for parsing of test results
- Configuration file control of test result parsing and report display.
- Supports multiple configuration files
- Generates HTML reports with highlighted pass/fail information
- Generates a text only report
- Provides easy to navigate Web Links to individual test result log files
- Tabulates Pass/Fail data for ease of use
- Email notification support via sendmail, smtp, or qmail
- Archiving of old results and maintaining of such on separate web page
- Deletion of old results
- Options to limit size of stored log files (runaway tests)

### 3 Requirements

RRT has the following Infrastructure and User Requirements:

- Perl 5.6.1 or Later
- A Unix based OS. This includes Cygwin
- Web browser
- During setup, an individual with a basic knowledge of Perl regular expressions
- Tar
- Gzip
- Sendmail, qmail, or smtp server (For the Email notification feature)
- Each test log file needs to have a unique name (Only, if not using the recursive mode)
- The log files that are to be parsed must have the following which can be uniquely identified with a Perl regular expression:
  1. Test Name
  2. Pass/Fail Info

### 4 Configuration

RRT is configured via a user defined configuration file that allows users to specify their unique environment. Users can setup email addresses, report formatting, search parameters, and environment info in the configuration file. RRT provides an example configuration file that can be used as a starting point. The following sections highlight some available configuration controls. A complete list with full details of each can be found in the RRT User Guide.

#### 4.1 Email Notification

Enables users to specify various options for controlling RRT's email function, including:

- Email Program to be used
  - sendmail
  - qmail
  - smtp
- smtp server
  - Domain where the smtp server lives, e.g. mail.usgov.gov
- Email Domain Name
  - Mechanism for appending a domain name to addresses that do not have one. This is useful when one is using a username as the address to send to.
- Email Addresses
  - Designation of whom will be sent Regression report related emails (failures, pointers to log files, etc.)

#### 4.2 Environment Configuration Parameters

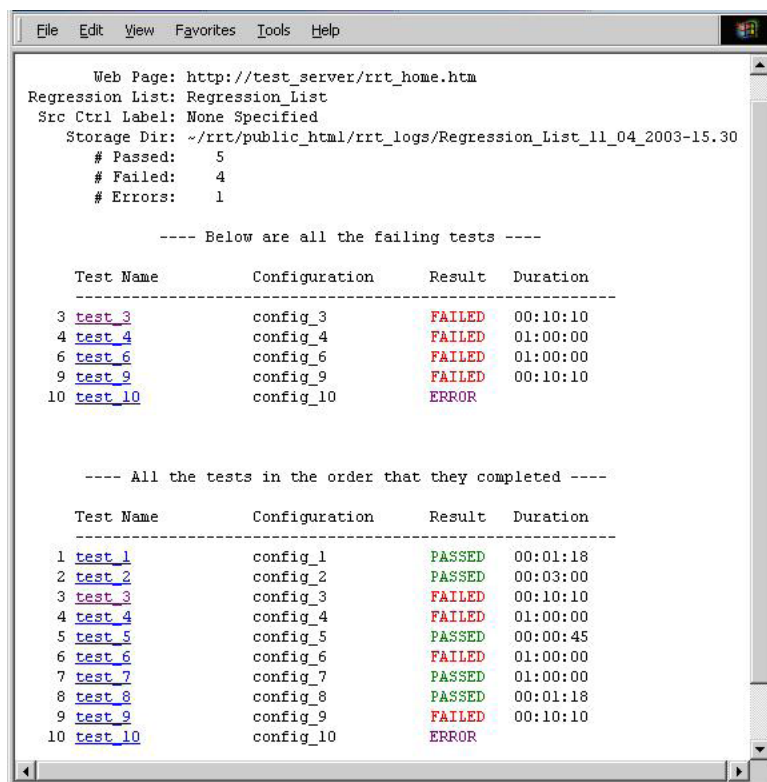
Used to designate such things as where the web pages, test logs, and reports are stored:

### 4.3 Performance Configuration Parameters

RRT provides several parameters to reduce the time it takes to search for, parse, and store log files. For example, users can parse just the beginning of the log file, look for runaway files and abort the parsing should a file size exceed a certain specified threshold, search only certain directories for results, etc.

### 4.4 Search Strings and Web Report Fields

Special parameters are provided that can be used to format the output information that is displayed on each of the report web pages. Users can define pass/fail search strings, column formatting, headers, etc. A sample regression report could be configured to display as below in Figure 4-1.



```

File Edit View Favorites Tools Help
Web Page: http://test_server/rrt_home.htm
Regression List: Regression_List
Src Ctrl Label: None Specified
Storage Dir: ~/rrt/public_html/rrt_logs/Regression_List_11_04_2003-15.30
# Passed: 5
# Failed: 4
# Errors: 1

---- Below are all the failing tests ----

Test Name      Configuration  Result  Duration
-----
3 test_3       config_3      FAILED  00:10:10
4 test_4       config_4      FAILED  01:00:00
6 test_6       config_6      FAILED  01:00:00
9 test_9       config_9      FAILED  00:10:10
10 test_10     config_10     ERROR

---- All the tests in the order that they completed ----

Test Name      Configuration  Result  Duration
-----
1 test_1       config_1      PASSED  00:01:18
2 test_2       config_2      PASSED  00:03:00
3 test_3       config_3      FAILED  00:10:10
4 test_4       config_4      FAILED  01:00:00
5 test_5       config_5      PASSED  00:00:45
6 test_6       config_6      FAILED  01:00:00
7 test_7       config_7      PASSED  01:00:00
8 test_8       config_8      PASSED  00:01:18
9 test_9       config_9      FAILED  00:10:10
10 test_10     config_10     ERROR

```

Figure 4-1: Sample Regression Report Page

## 5 Command Line Options

RRT also includes a number of command line options for controlling the regression reporting. The following sections highlight some available command line options. A complete list with full details of each can be found in the RRT User Guide.

### 5.1 Address Override

This option is provided as a means to override the email addresses that are set within the configuration file and to send an email notification to one or more individuals. A typical application is where a common

configuration file is used for parsing data; however, the parsing of test results is for a particular individual and not the entire team. Can use “multiple –address” options to specify multiple people.

## 5.2 Archiving

As time goes by the master web page can become cluttered with many lines that contain links to information that is of little interest. However, it is not desirable to remove the information completely. An archive option is provided that will tar & compress the report log files, remove the links from the report files to the individual test logs, and move the report links from the home page to an archive page. See Figure 5-1: The Archive Web Page below:



**Figure 5-1: The Archive Web Page**

## 5.3 Configuration Files

The configuration file switch (`-config_file`) is provided to allow one to maintain multiple config files. This provides one with the option of maintaining multiple report types, web pages, email lists, etc.

## 5.4 Date Stamps

The `-date_stamp` option is used for two purposes. One, as a point of reference to the time the regression results were generated. Two, as a means to uniquely store results that have the same regression list name. The RRT default is to generate a date stamp of the form `MM_DD_YYYY-hh.mm` at the time the tool is called. However, if one would like to use a different unique labeling system one can override this behavior by supplying one's own date stamp.

### 5.5 **Deleting Old Archives**

After a certain period of time, the archived results that are being maintained are no longer needed. A `delete_archive` option has been provided that will remove the archived test logs (tar.gz). The associated report web pages will be maintained. In order to remove the reports from the web pages use the `delete_reports` option in conjunction with the `-delete_archives` option. Just like in the archive option this command takes as an argument the number of weeks to keep.

### 5.6 **Deleting Old Reports**

After a certain period of time, the report results that are being maintained on the archived web page are no longer needed. A `-delete_reports` option has been provided that will remove the report files and the report links from the effected web pages.

### 5.7 **Email**

One of the main benefits of RRT is to allow an entire group to have quick access to the regression results. The Email feature is provided to notify the appropriate individuals when there is new data for review. The address list is maintained in the configuration file or by specifying the `-address <email_address>` option on the command line

### 5.8 **Force**

Perform the deletion of archived results without a confirmation prompt. This is provided for the scenario when one has automated mechanism for deleting old archives, such as with cron.

### 5.9 **List Name**

An option is provided to allow one to label the regression results with a user defined name. This is typically the name of the regression list. The default is to label the results "Regression\_list"

### 5.10 **No Web Page**

The `-no_web_page` option disables the update/creation of the report web pages and prints the regression report to STDOUT.

### 5.11 **Parse Only**

Parse the test results and return pass/fail data only. Do not store the data, update the web pages, or send email notifications. The output is displayed to STDOUT.

### 5.12 **Recursive**

To parse the input directory recursively use the `-recursive` switch. This allows for a complex test environment and for test names to have the same file name. The uniqueness is in the path name. The first command line argument must be a directory. Any other arguments following this will be ignored.

### 5.13 **Source Control Labels**

A switch is provided to enable a method to specify the source control label/tag that is needed to reproduce the regression environment.

## 6 **Company Background**

Paradigm Works is a leading chip design and verification services company. The company is recognized for engineering excellence, integrity in business, and overall productivity and cost effectiveness.

We provide expert consultants and contractors both on site and offshore to assist in complex chip developments. We offer world class domain expertise (PCI Express, USB, Ethernet), application knowledge (Networking, Computing, Storage, Wireless), and leverage Paradigm Works suite of productivity accelerator software (VerificationWorks™, ReleaseWorks®, and SystemVerilog FrameWorks™) to help clients bring their innovations to market as quickly as possible.

For more information on Paradigm Works products and services call 978-824-1400 or see our web site at [www.paradigm-works.com](http://www.paradigm-works.com).