Test Station Data Log Requirements

Rev 1.1a

|  |  |
| --- | --- |
| Last Updated: | October 24th, 2015 |
| Author: | Wei Wang |

|  |  |  |
| --- | --- | --- |
| **Revision History** | | |
| Time | Author | Comment |
| Feb 20th, 2012 | Wei Wang | Initial Draft |
| Mar 6th, 2012 | Wei Wang | Added the details of csv log. Added ToC |
| Mar 20th, 2012 | Wei Wang | Added alignment requirement for debug logs  Added spec for Pass/fail tests  Added spec for tester location without DUT  Upped the version to 1.0a |
| October 8th, 2014 | Wei Wang | Added format for result and limits in flow log  Removed the requirement for DiagsOverUSB log  Added test id requirement in csv log  Updated the timestamp format  Version 1.1 |
| Oct 24th, 2015 | Martin Zhang | Added requirement ID |

The Logs 4

Naming the Tests 4

Debug Logs 4

1. Comment Elements 5

Time stamps 5

Comments 5

Alignment 5

2. UART Log 5

Content 5

File Name 6

3. Test Flow Log 6

Content 6

File Name 8

Data Log (CSV log) 9

Content 9

File Name 10

# The Logs

[DLR-01] The station software is required to generate the logs in Table 1 and upload them to PDCA. Not all logs are required for all units all the time, but the software needs to be capable of turning on/off the generation of any of these logs without recompilation, even in production.

[DLR-02] In Production, if the test result is FAIL, then the all the logs should be turned on and uploaded to PDCA for retest.

Table 1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Log | Content | One per DUT | Required in Engineering Build | Required in Production if passed |
| UART Log | Output from the DUT’s UART | Y | √ | X |
| Test Flow Log | Tester actions interspersed with DUT’s UART output | Y | √ | X |
| Csv log | Parametric data for each device | N | √ | X |

The details of what should be included in each log are described in the rest of the document.

[DLR-03]

# Naming the Tests

1. Any character can be used in the test names except for comma, because the test names will be used in the csv file.
2. There is no limit on how long the test names can be. And it’s OK to have spaces in the test name
3. Each test must have a unique test id. This test id is a convenience for parser writers; therefore although the test names are encouraged to be descriptive, the test id should be short. The test id must be one word so no space is allowed.

# Debug Logs

[DLR-04] The first three logs are generated for each DUT. Their main purpose is to help debug test flow issues and FA.

It’s imperative the formats of the logs follow the specification in this document to facilitate automatic parsing and interpretation.

## Comment Elements

### Time stamps

[DLR-05] In these logs, all time stamps should have the same format:

1. Begins at the beginning of a line.
2. Format is “yyyy/mm/dd hh:mm:ss:ddddd”.
3. Between the timestamp and the actual log content on the same line, there should be and only be spaces. No comma, colon or any other delineators.
4. Not every line is required to have a timestamp. Sometimes the output from the DUT has line changes in it. It’s OK to not insert timestamps in every line in this case.

### Comments

[DLR-06] Vendors can add contents in the logs to facilitate their own debugging. These should appear as C style comments in the logs. In other words, two forms of comments are allowed:

1. Single-line comments: Everything after “//” until the end of line.
2. Multi-line comments: Everything between “/\*” and “\*/”

### Alignment

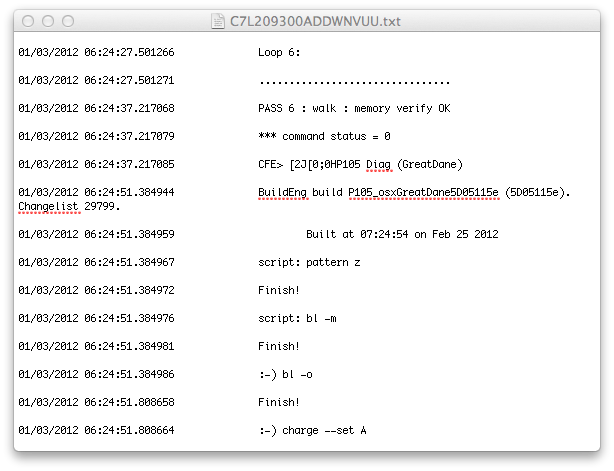
In these logs some lines begin with a time stamp and some lines don’t. On the lines that don’t begin with a time stamp, the logging software should add indentation so the content of all lines are aligned.

The logging software needs to look for line breaks in the response returned from diags to make sure indentation are added even though there is no line change in the logging code.

## UART Log

### Content

This log needs to capture all the output from DUT’s UART during test. This include diags commands and their responses, iBoot and Diags boot message, memory test outputs, etc. The FCT software should insert time stamp at the beginning of each write to the log file. A sample log looks like Figure 1.

Figure 1

### File Name

The name of this log should be in this format: [SerialNumber]\_[station id]\_[site id]\_ [timestamp]\_uart.txt

* Serial Number is the SN of the DUT
* Timestamp should be of the format: mm-dd-hh-mm-ss
* There should be no “\_” in the station id
* Site id is the location of the particular fixture. In a 4-up fixture, site id can be 1 through 4.

## Test Flow Log

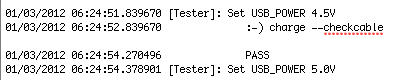
### Content

The test flow log includes all the content of the UART log, but it also includes these extra contents:

#### Tester actions that happen between DUT reactions.

For instance, Figure 2 shows how the USB bus valid check should be recorded:

Figure 2



Note that:

1. Each tester action line needs to have a timestamp.
2. It’s marked with the “[Tester]:” label.
3. There is at least one space between the colon after “[Tester]” and the rest of the line.
4. There is no syntax requirement for actual action for now. However, anything that touches the DUT should be recorded. Examples include:

* A relay connected directly to a test point is closed/opened.
* Voltage/Current stimuli to the DUT are changed.
* Measurement of the DUT’s voltage/current is performed.

This flow log is only required for the functional test part of the test program. It’s not required for Open/Short, MDA and Boundary scan.

#### Markers for beginning of Test and SubTest

The log for the same test above with the markers include should look like Figure 3:

Figure 3

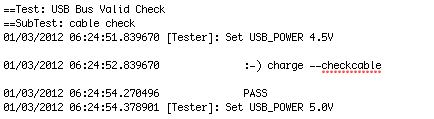


Figure 4 shows how this looks like in the corresponding test plan:

Figure 4

Macintosh HD:Users:wei_wang:Desktop:Screen Shot 2012-03-06 at 5.55.53 PM.png

Note that:

1. The marker lines don’t need timestamps
2. The marker lines begin with “==Test:” or “==SubTest:”
3. The test name and subtest name should be consistent with the test plan. At least one of them need to include the unique test id.

#### Test Limits

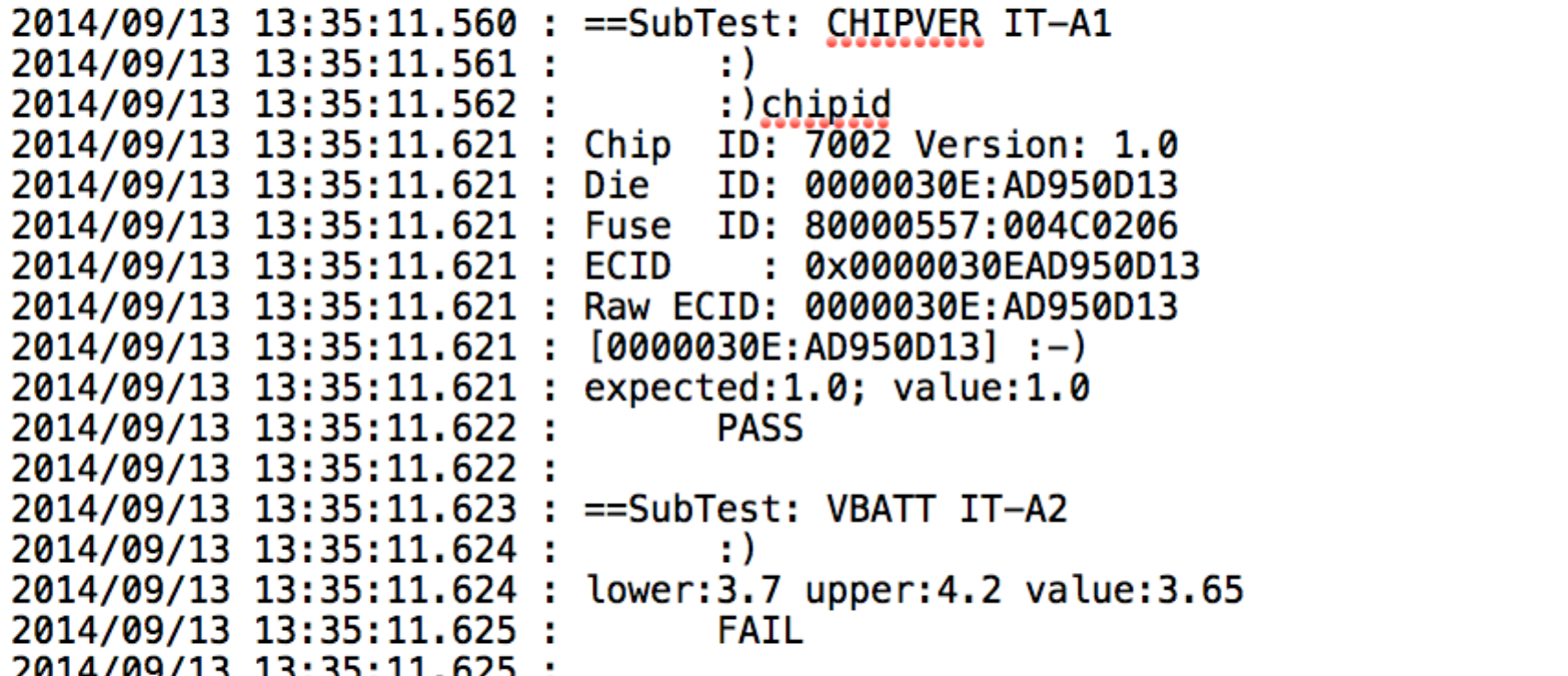
In each subtest section, there should be a separate line that shows the test limits for this subtest.

* If the test is numerical and have upper and lower limits, this line should have this format: “lower:[low limit]; higher:[high limit] ;value:[value]”.
* If the test is only pass/fail with certain expected values, this line should have this format: “expected:[all possible values separated by | ]; value:[value]”
* This line doesn’t need to have a timestamp. It doesn’t matter if this line is indent or not, as long as it’s on a separate line.

#### Test Results

Following the line for the test limits there should be a line that indicates if this test has passed or failed. This should be a separate line that only has one of two possible values “PASS” or “FAIL”. It’s not case sensitive.

Below is a sample subtest section:



### File Name

The name of this log should be in this format: [SerialNumber]\_[station id]\_[site id]\_[timestamp]\_flow.txt

# Data Log (CSV log)

### Content

#### Comment:

Only C-style single line comments are allowed. However, these are not intended as comments in the data lines. They are intended for vendors to insert vendor specific headers.

#### Header line:

* The first line should record the vendor name, test software version and the test script version, separated by commas. There is no specific requirements for the syntax of test software version and test script version, only that there is no comma in the version name. A sample looks like: TRI, 4P56, P105\_V20110312
* On the first row lining up with each test name on the second row, there should be a unique test id for each test item. This unique test id can contain any character except for space, tab or comma.

#### Data Headers:

Because this is a csv file, the delineators are commas. It’s important that there is no comma in any test names.

1. The first row is the header line.
2. The second row should include these columns in order:

* Serial Number
* Config
* Station id
* Site id
* Pass/Fail
* Error Message
* Failed list
* Test start time
* Total Test time
* Open/Short (pass/fail only)
* MDA (pass/fail only)
* Rail short tests (pass/fail only)
* All functional test names in order (these names should line up with the unique test id on the header line)
* All Rail short test names in order (these names should line up with the unique test id on the header line)
* All MDA test names in order (these names should line up with the unique test id on the header line)

1. The third row should have the upper limits for all tests lined up with their test names. For non-parametric tests the limits should be N/A.
2. The fourth row should have the lower limits for all tests lined up with their test names. For non-parametric tests the limits should be N/A.
3. The fifth row should have the units for all tests lined up with their test names. For non-parametric tests the unit should be N/A.
4. Beginning on the sixth row, each row should record one DUT’s data, with each datum lined up with the test names. Make sure there is no comma in any data.

* For tests that are skipped, the datum should be “Skipped”
* For tests that are not performed because the DUT stopped on fail, the datum should be empty.
* For tests that expect a certain value, the datum should be the actual value, not “Pass” of “Fail”
* For tests whose results are only pass or fail, the datum should be “PASS” or “FAIL” only, with no other description.
* If there is no DUT in a site, no data should be recorded for that row. There should not be a whole row of “Skipped”.

### File Name

There are not requirements for the actual file name. But a new csv file should be generated whenever the test script is changed.