

stdf2xls4 User Manual
February 26, 2017

Revision History

Version	Author	Date	Changes
1.0	Eric West	2/26/17	Update to stdf2xls4 version 4.1.2

Contents

1	Introduction	3
2	Getting the program	3
3	Installation	3
4	Usage	3
4.1	Generating a Spreadsheet	3
4.2	Converting STDF to ASCII	3
4.3	Getting a Summary of Options	3
5	Command Line Options	4
6	Support for Alphanumeric Serial Numbers	6
7	Support for Generic Data per Device	6
8	Header Fields	6
8.1	Legacy Header Fields	6
9	Filename Timestamps	7

1 Introduction

stdf2xls4 is a java 8 program that converts STDF files into spreadsheet form. A couple of major features are that it correctly handles duplicate test names and test numbers, and it can handle dynamic limits - test limits that are dependent on previous test results. The program will run on any platform that supports java 8.

2 Getting the program

The code is available on github as an eclipse project. To get the executable jar file clone the repository with the following command:

```
git clone https://github.com/makechip/stdf2xls4
```

The jar file is in the stdf2xls4/dist directory.

To compile the source code, import the project into eclipse as a git project.

3 Installation

Installation is just a matter of possibly copying the jar file to a more convenient directory. Optionally a script can be made to actually run the jar file if desired.

4 Usage

4.1 Generating a Spreadsheet

```
java [-mx<M>] -jar <path_to_std2xls4_jar_file> [options] -x <spreadhseet_file> <stdf_files>
```

- <M> is the amount of memory to allocate to the java virtual machine for example, 4G, 500M, 12G.

options are explained in the next section, and are optional.

- The spreadsheet_file must have a suffix of ".xls" or ".xlsx". The spreadsheet format is determined from this suffix.
- There can be one STDF file or many STDF files specified; also "*.std" or "*.stdf" will work.

4.2 Converting STDF to ASCII

```
java -jar <path_to_std2xls4_jar_file> -d <stdf_files>
```

4.3 Getting a Summary of Options

```
java -jar <path_to_std2xls4_jar_file> --help
```

5 Command Line Options

Options can be specified as a single dash followed by a single character, or a double dash followed by multiple characters.

- **-a [<char>] or --pin-suffix [<char>]**
This option tells the program that if a test name of a ParametricTestRecord has a character <char> in it then the characters following <char> are to be interpreted as a pin name for that test. If <char> is not specified, then by default '@' will be used. The pin name is displayed on a separate row or column in the test header.
- **-b or --one-page**
Normally STDF files from different wafers or test steps are displayed in different spreadsheets. This option will combine them all on one page with an additional row or column for the test step or wafer number.
- **-c or --columns**
Specifying this option will increase the maximum number of columns in the spreadsheet from 1024 to 16384. Note that with this option the spreadsheet will not be readable with Libreoffice.
- **-d or --dump**
This option causes the STDF data to be dumped in ASCII format to STDOUT.
- **-g or --gui**
This option will run the program from a GUI instead of the command line. This option is experimental and will probably be deprecated in a future version.
- **-h or --help**
Prints a summary of the command line options.
- **-j or --jxl-xls-name**
This option specified to use the JExcel library instead of the Apache POI library for spreadsheet generation. Sometimes the Apache library for "xls" format is problematic. The JExcel library can be tried to see if it resolves any formatting problems. Note that this option has no effect when using "xlsx" or xml format, and xml format is recommended anyways.
- **-m or --modifier**
This is an experimental feature that can modify STDF fields on the fly as the STDF is loaded. Currently it is only enabled for DatalogTextRecords. This option has the form of:
-m "R:Record_t F:FieldDescriptor C:Condition_t V:oldValue N:newValue"
Where record type currently can only be DTR, the field descriptor can only be TEXT_DAT, and Condition_t can be EQUALS, CONTAINS, or TRUE. Also, multiple modifiers may be specified by using multiple -m options. This implementation is lame, and will probably later be replaced by a regular expression engine.

```
-m "R:DTR F:TEXT_DAT C:CONTAINS V:2.00 N:2.0"
```

This will check every DatalogTextRecord for the string "2.00", and if found, it will replace it with "2.0". Currently whitespace in the string cannot be handled. The Condition TRUE is currently useless. The EQUALS condition requires that the entire text field be equal to the oldValue for a replacement to occur.

- **-n or --no-wrap-test-names**
When the the test names run horizontally across the spreadsheet they will be wrapped to keep the columns from getting too wide. This option will prevent wrapping the test names, but can give very wide columns.
- **-o or --no-overwrite**
Currently this option is not working.
- **-p <integer> or --precision <integer>**
This option specifies how many digits will be to the right of the decimal point for floating point values. The default is 3.
- **-r or --rotate**
This option transposes the spreadsheet. By default test names run horizontally across the spreadsheet and devices run vertically down the spreadsheet. With the -r option the test names will run vertically down the spreadsheet and devices will run horizontally across the spreadsheet.
- **-s or --sort**
this options sorts the data by serial number or XY-coordinate (for wafersort).
- **-v or --dont-skip-search-fails**
Verigy (AKA Advantest) testers can output bogus test data as tests related to searches. By default these will be ignored. This option will include them in the spreadsheet.
- **-x <file.xls[x]> or --xls-name <file.xlsx[x]>**
This options specifies the name of the spreadsheet file and is required (unless using the -d option). If the suffix of the filename is ".xlsx" then the newer xml format will be used for the spreadsheet. Otherwise the older ".xls" format will be used. In the futute the xls format may be deprecated.
- **-y or --dynamic limits**
If this option is used, then the high and low limits of all parametric tests are searched, and if they vary by more than 0.1 % then the corresponding test will use dynamic limit which means that the column to the left of the test will be the lower limit used and the column to the right of the test will be the upper limit used.

6 Support for Alphanumeric Serial Numbers

For alpha-numeric serial numbers, print a text field to the datalog at the beginning of the test flow that looks like this:

```
TEXT_DATA: S/N : <serial_number>
```

The strings "TEXT_DATA" and "S/N" must match exactly what is shown here. Whitespace around the colons is ignored. The maximum length of any string printed to the datalogger must not exceed 255 characters.

7 Support for Generic Data per Device

Arbitrary data for each device may be mapped to the spreadsheet as a test result along with the rest of the test data. To get this information into the STDF file, print a text field to the datalogger using the format:

```
TEXT_DATA : test_name : value [(units)] [ : test_number [ : site_number [ : head_number ]]]
```

All fields after the value are optional. Whitespace around the colons is ignored.

8 Header Fields

It is often desirable to include header information in each spreadsheet. The header information can be encoded into datalog text fields that are printed out for each device. Note that it is expected that the header data be constant for an entire test session. If the header information changes within a set of STDF files, then there will be multiple spreadsheets generated (within a workbook) with each spreadsheet having different header information.

To generate a header field, print the field to the datalogger using this format:

```
>>> <header_name> : <header_value>
```

The prefix ">>>" indicates a header field. The header name can be any string (not containing a colon) and the header value can be any string.

8.1 Legacy Header Fields

There were a set of fixed header fields that are still supported, but are now deprecated (don't use them on new test programs). These field were simply a header name followed by a header value. The recognized header names are:

- CUSTOMER:
- DEVICE NUMBER:
- SOW:
- CUSTOMER PO#:
- TESTER:
- TEST PROGRAM:
- LOADBOARD:
- CONTROL SERIAL #s:
- JOB #:
- LOT #:
- STEP #:
- TEMPERATURE:

These header names may be combined with the new header format if necessary.

9 Filename Timestamps

Filenames conforming to a special timestamp format will be automatically detected. If there is one device per STDF file, and the STDF filename uses this format:

```
<any_string>_YYYYMMDDhhmmss.std[f]
```

Then the timestamp will be included in the spreadsheet along with each serial number.