



stdf2xls5 User Manual
February 26, 2020

Revision History

| Version | Author | Date | Changes |
|---------|-----------|---------|-----------------|
| 1.0 | Eric West | 2/26/20 | Initial Release |

Contents

| | | |
|-----------|--|-----------|
| 1 | Introduction | 3 |
| 2 | Getting the program | 3 |
| 3 | Installation | 3 |
| 3.1 | GNU/Linux Installation | 3 |
| 3.2 | Windows Installation | 3 |
| 4 | Usage | 4 |
| 4.1 | -s or --genSpreadsheets | 5 |
| 4.2 | -h or --genHistograms | 10 |
| 4.3 | -w or --genWafermaps | 10 |
| 4.4 | -d or --dumptext | 10 |
| 4.5 | -b or --dumpBytes | 10 |
| 4.6 | -D or --digest | 10 |
| 4.7 | -m or --modify | 10 |
| 4.8 | -g or --generateRCFile | 10 |
| 4.9 | -o or --outputDir | 10 |
| 5 | Command Line Options | 10 |
| 6 | Support for Alphanumeric Serial Numbers | 12 |
| 7 | Custom Spreadsheet Data Values | 12 |
| 8 | Spreadsheet Header Fields | 12 |
| 8.1 | Legacy Header Fields | 13 |
| 9 | Modifying STDF Text fields | 13 |
| 10 | Wafermaps | 13 |
| 11 | Histograms | 13 |
| 12 | Look and Feel Customization | 13 |

1 Introduction

stdf2xls 5.0 is a program that converts STDF files into spreadsheets, wafermaps, and/or histograms. It is natively compiled from the D-language and therefore it uses much less memory than the previous stdf2xls 4.0 java program, and is significantly faster too. It has many new features:

- Spreadsheet and ASCII wafermaps
- Spreadsheet histograms
- Ability to display hex, integer, and string data values in the spreadsheet
- New algorithm correctly orders tests even if the testflow varies from device to device
- Spreadsheet colors and fonts are now customizable
- Improved spreadsheet layout
- Many ways to sort and order device data
- Ability to modify any STDF text field with regular expressions
- Ability to write out STDF files
- Multiple different devices types can be processed simultaneously

2 Getting the program

The source code is available on github. The source code can be downloaded from github with git, or just the executable for Windows or GNU/Linux may be downloaded from the github web page. This manual may also be copied from the github webpage.

To download the software with git use the following command:

```
git clone https://github.com/itestinc/stdf2xls.git
```

To compile the source code on GNU/Linux you also need dub, gcc and dmd (dmd is the D compiler).

3 Installation

To just install the executable, the program can be downloaded from github.com.

3.1 GNU/Linux Installation

For GNU/Linux go to <https://github.com/itestinc/stdf2xls>, click on the dist folder, then click on the stdf2xls file, then press the Download button. After the file has downloaded put it in a directory that in in your executable search path.

3.2 Windows Installation

For Windows go to <https://github.com/itestinc/stdf2xls>, click on the dist folder, then click on the stdf2xls.exe file, then press the Download button. After the file has downloaded put it in a directory that in in your executable search path.

4 Usage

A summary of the command line options can be printed by running the program with the '--help' option. More detailed information about the command line options is given in this manual.

```
stdf2xls --help
```

gives the following output:

```
Options:
-a      --extract-pin Extract pin name from test name suffix (default delimiter = '@')
-b      --dumpBytes dump the STDF in ascii byte form
-d      --dumptext dump the STDF in text form
-m      --modify modify a string field in specified record type.
        Example: -m 'MIR TST\_TEMP "TEMPERATURE :"' "TEMPERATURE:":'
-o      --outputDir write out the STDF to this directory. Specifying this will cause the STDF to be written back out.
-p      --pin-delimiter Delimiter character that separates pin name from test name (Default = '@')
-i      --ignoreSerialMarker Ignore the serial marker and use STDF part ID instead
-D      --digest Summarize file contents
-s      --genSpreadsheets Generate spreadsheet(s)
-S      --so Spreadsheet output filename(s); name may contain variables for device, and/or lot
Default = %device%_%lot%.xlsx
-r      --rotate Transpose spreadsheet so there is one device per column instead of one device per row
--sortType Sort devices by alphanumeric serial number, then by time. See the manual for valid sort types
-c      --1kcol limit to 1000 columns for libreoffice - default is 16360 columns
-Y      --noDynamicLimits Don't check for and show dynamic limits
-w      --genWafermaps Generate wafer map(s)
-W      --wo Wafermap output filename(s); name may contain variables for device, wafer, and/or lot
Default = %device%_%lot%_%wafer%.xlsx
-P      --pattern fill wafermap bins with patterns instead of color
-A      --dumpAscii dump the wafer map in ASCII form
-n      --notch Rotate the wafer map for desired notch position: top|bottom|left|right.
-h      --genHistograms Generate histogram(s)
-H      --ho Histogram output filename(s); name may contain variables for device, step, lot, and/or testID
Default = %device%_histograms.pdf
--binCategory Specify if bins should be divided by SITE, LOT, TEMPerature or NONE. Default = NONE
Note: if --ho contains %lot% then dividing bins by lot does not make sense
-g      --generateRCFile Generate a default ".stdf2xlsrxc" file
-t      --channel-type Channel type: AUTO, CHANNEL, PHYSICAL, or LOGICAL. Only use this if you know what you are doing.
-v      --verbose Verbosity level. Default is 1 which means print only warnings. 0 means don't print anything
-V      --verify Verify written STDF; only useful if \texttt{-{}}-outputDir is specified. For testing purposes only.
--noIgnoreMiscHeader Don't ignore custom user header items when comparing headers from different STDF files
-h      --help This help information.
```

There are nine primary operations that can be done with the program, and they may be done individually or simultaneously:

- -s or --genSpreadsheets will generate spreadsheets for the STDF measurement data
- -h or --genHistograms will generate histograms for the STDF measurement data
- -w or --genWafermaps will generate wafermaps for the STDF bin data
- -d or --dumptext will generate an ASCII dump of the STDF file(s)
- -b or --dumpBytes will generate an ASCII dump of all of the bytes in the STDF file(s)
- -D or --digest will dump each unique set of header information found in the STDF file(s)
- -m or --modify '<record> <field> <fromRegex>: <toRegex>' will modify any text field in any record type
- -g or --generateRCFile will generate an "rc" file in the home directory called ".stdf2xlsrxc" which contains all the default color, font, and logo settings.
- -o or --outputDir <folder_name> will cause the STDF files that are read in to be written out to this folder

Each of these major options have several more options to refine their behavior, and their usage and options will be discussed in the following sections.

4.1 -s or --genSpreadsheets

Spreadsheet generation has nine options that control aspects of how the spreadsheet is generated. You must give the -s or --genSpreadsheet option to get a datalog spreadsheet.

- -a --extract-pin Extract pin name from test name suffix (default delimiter = '@')
- -p --pin-delimiter Delimiter character that separates pin name from test name (Default = '@')
- -i --ignoreSerialMarker Ignore the serial marker and use STDF part ID instead
- -S --so Spreadsheet output filename(s); name may contain variables for device, and/or lot
- -r --rotate Transpose spreadsheet so there is one device per column instead of one device per row
- -c --1kcol limit to 1000 columns for libreoffice - default is 16360 columns
- -Y --noDynamicLimits Don't check for and show dynamic limits
- -t --channel-type Channel type: AUTO, CHANNEL, PHYSICAL, or LOGICAL.
- -m --modify modify a string field in specified record type.

4.1.1 -a or --extract-pin

Some testers like the Advantest 93K will append a pin name on to the end of a test name to indicate the pin that is being tested. By default it uses an '@' to delimit the test name from the pin name. For example "myTestName@VCC". This option will remove the delimiter and pin name from the test name, and put the pin name in the pin column or row of the spreadsheet. You may use any character for the deliter, but '@' is used by default. See next section.

4.1.2 -p or --pinDelimiter

This option allows you to specify the delimiter that is used to separate the test name and pin name. It is only necessary to use this option if the delimiter is not '@'.

4.1.3 -i or --ignoreSerialMarker

There are two ways to specify a serial number for a device. If you do nothing then stdf2xls will use the PART_ID field in the PRR record as the serial number. This field is set by the tester automatically. However if you want to assign your own alpha-numeric serial numbers you can put them into a Datalog Text Record using a special format. To put your own serial numbers in to the STDF, print a string to the datalogger using this format:

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

For example,

```
TEXT_DATA : S/N : A1
```

On the Advantest 93K a Datalog Text Record is generated by using the PUT_DATALOG() function. The stdf2xls program will prioritize the Datalog Text Record serial marker over the PART_ID field in the PRR record. If you want to prioritize this PART_ID field over the Datalog Text Record serial marker, then use this option. Generally you won't want to use this option.

4.1.4 -S or -so

This option is used to specify the spreadsheet filename. It is not necessary to use this option. By default the spreadsheet filename will be <deviceName>_<lot_id>.xlsx. Note that you can process multiple device types simultaneously and each device and lot will be sent to a different spreadsheet file. With this option you can specify the datalog spreadsheet filename, and you can use variables to specify the lot and/or device in the filename. For example:

```
-S device_%device%_lot_%lot%.xlsx
```

If your device was 8087 and your lot number was N5432S, then this would give filename of "device_8087_lot_N5432S.xlsx"

Note that if the actual lot or device number contains a '/' character, then the '/' will be replaced with a '%' character because you can't have slashes in a filename. If the lot or device number contains a space character, then the space will be replaced with a '_' character. This is because spaces in filenames are evil.

4.1.5 -r or -rotate

By default the datalog spreadsheet is generated with tests in columns and devices in rows. This option will transpose the spreadsheet so that devices are in columns and tests are in rows.

4.1.6 -c or -1kcol

Use this option if you are using libreoffice or openoffice.org. This limits the number of columns to 1000. Libreoffice and openoffice.org will truncate any data beyond 1024 columns. By default 32K columns will be used for MS Excel. In any case if the number of columns exceeds these limits then the data will be continued on another tab in the workbook.

4.1.7 -Y or -noDynamicLimits

By default all parametric tests are scanned for non-constant limits. If non-constant limits are detected for certain tests, then the spreadsheet is formatted differently for those tests such that each test result is surrounded by the limits used for that test. If you don't want this behavior then use this option, but realize the limits will not be accurate.

4.1.8 -t or -channel-type

For Multiple Parametric Test Records the pin information is obtained from the Pin Map Records which map a pin index number to a pin name. Unfortunately the Pin Map Record has three fields where a pin name *might* be stored. Most of the time stdf2xls will use the correct field to get the correct pin name, but occasionally a new tester may use the wrong field. In that case this option can be used to force stdf2xls to use the correct field. If your pin names are not coming out correctly for Multiple Parametric Test Result records, then do an ASCII dump of the STDF file, and look at the Pin Map Records, and it should be obvious which field to specify with this option.

4.1.9 -m or -modify

This option can be used to modify a string field in any STDF record. It uses a powerful regular-expression engine that is documented at <http://www.regular-expressions.info/>. This option has the form:

```
modify '<record> <field> \"<fromRegex>\" \"<toRegex>\"'
```

Note that the single quotes and double quotes are needed so that the shell interprets the command line correctly. Note that you can also specify this option multiple times on the command line to do several edits simultaneously.

The <record> parameter is the three-letter record name as it is specified in the STDF specification. The <field> parameter is the field name as given in the STDF specification. The valid record and field names that may be modified are shown below:

- Record ATR
 - Field Name: "CMD_LINE"
- Record BPS
 - Field Name: "SEQ_NAME"
- Record DTR
 - Field Name: "TEXT_DAT"
- Record FTR
 - Field Name: "VECT_NAM"
 - Field Name: "TIME_SET"
 - Field Name: "OP_CODE"
 - Field Name: "TEST_TXT"
 - Field Name: "ALARM_ID"
 - Field Name: "PROG_TXT"
 - Field Name: "RSLT_TXT"
- Record HBR
 - Field Name: "HBIN_NAM"
- Record MIR
 - Field Name: "LOT_ID"
 - Field Name: "PART_TYP"
 - Field Name: "NODE_NAM"
 - Field Name: "TSTR_TYP"
 - Field Name: "JOB_NAM"
 - Field Name: "JOB_REV"
 - Field Name: "SBLOT_ID"
 - Field Name: "OPER_NAM"
 - Field Name: "EXEC_TYP"
 - Field Name: "EXEC_VER"
 - Field Name: "TEST_COD"
 - Field Name: "TST_TEMP"
 - Field Name: "USER_TXT"
 - Field Name: "AUX_FILE"

- Field Name: "PKG_TYP"
- Field Name: "FAMILY_ID"
- Field Name: "DATE_COD"
- Field Name: "FACIL_ID"
- Field Name: "FLOOR_ID"
- Field Name: "PROC_ID"
- Field Name: "OPER_FRQ"
- Field Name: "SPEC_NAM"
- Field Name: "SPEC_VER"
- Field Name: "FLOW_ID"
- Field Name: "SETUP_ID"
- Field Name: "DSGN_REV"
- Field Name: "ENG_ID"
- Field Name: "ROM_COD"
- Field Name: "SERL_NUM"
- Field Name: "SUPR_NAM"
- Record MPR
 - Field Name: "TEST_TXT"
 - Field Name: "ALARM_ID"
 - Field Name: "UNITS"
 - Field Name: "C_RESFMT"
 - Field Name: "C_LLMFMT"
 - Field Name: "UNITS_IN"
 - Field Name: "C_HLMFMT"
- Record MRR
 - Field Name: "USR_DESC"
 - Field Name: "EXC_DESC"
- Record PGR
 - Field Name: "GRP_NAME"
- Record PLR
 - Field Name: "PGM_CHAR"
 - Field Name: "RTN_CHAR"
 - Field Name: "PGM_CHAL"
 - Field Name: "RTN_CHAL"
- Record PMR
 - Field Name: "CHAN_NAM"
 - Field Name: "PHY_NAM"

- Field Name: "LOG_NAM"
- Record PRR
 - Field Name: "PART_ID"
 - Field Name: "PART_TXT"
- Record PTR
 - Field Name: "TEST_TXT"
 - Field Name: "ALARM_ID"
 - Field Name: "UNITS"
 - Field Name: "C_RESFMT"
 - Field Name: "C_LLMFMT"
 - Field Name: "C_HLMFMT"
- Record SBR
 - Field Name: "SBIN_NAM"
- Record SDR
 - Field Name: "HAND_TYP"
 - Field Name: "HAND_ID"
 - Field Name: "CARD_TYP"
 - Field Name: "CARD_ID"
 - Field Name: "LOAD_TYP"
 - Field Name: "LOAD_ID"
 - Field Name: "DIB_TYP"
 - Field Name: "DIB_ID"
 - Field Name: "CABL_TYP"
 - Field Name: "CABL_ID"
 - Field Name: "CONT_TYP"
 - Field Name: "CONT_ID"
 - Field Name: "LASR_TYP"
 - Field Name: "LASR_ID"
 - Field Name: "EXTR_TYP"
 - Field Name: "EXTR_ID"
- Record TSR
 - Field Name: "TEST_NAM"
 - Field Name: "SEQ_NAME"
 - Field Name: "TEST_LBL"
- Record WIR
 - Field Name: "WAFER_ID"

- Record WRR
 - Field Name: "WAFER_ID"
 - Field Name: "FABWF_ID"
 - Field Name: "FRAME_ID"
 - Field Name: "MASK_ID"
 - Field Name: "USR_DESC"
 - Field Name: "EXC_DESC"

4.2 -h or --genHistograms

4.3 -w or --genWafermaps

4.4 -d or --dumptext

4.5 -b or --dumpBytes

4.6 -D or --digest

4.7 -m or --modify

4.8 -g or --generateRCFile

4.9 -o or --outputDir

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.
- -l <file> or --logo <file>
This option specifies a logo (PNG file) to include in the upper left corner of the spreadsheet. The logo should have an aspect ratio of about 290(W) to 120(H). It will be scaled to fit within the required area.
- -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.
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 future 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 Custom Spreadsheet Data Values

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 Spreadsheet 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 Modifying STDF Text fields

10 Wafermaps

11 Histograms

12 Look and Feel Customization