Artwork Conversion Software, Inc.

WMBatch/WMLib Commands

Overview

WMBatch/WMLib can be controlled via an ASCII command file. The command file contains commands recognized by the application to perform parsing, conversion, manipulation and writing of various wafer map formats. This document describes the available keywords supported and the syntax for controlling WMBatch/WMLib.

WMBatch can execute a command file using the built-in command line parameter:

wmbatch64.exe -command_file:<filename>

Comments

```
# WMBatch Command File
```

Comments can be made to annotate commands in the command file. Comments are recognized by placing a # character as the first non-whitespace character on a line.

Quotes

Expected command parameters should be enclosed inside double quotes when they contain whitespace, such as long file names, bin descriptions, etc.

Input Directory

```
input_dir “<dirname>”
```

Define a common input directory so that all subsequent input file names are to be opened from the defined input directory. Only unqualified input file names will be opened from the input directory. Fully qualified path names for input files will override this command.

Output Directory

```
output_dir "<dirname>"
```

Define a common output directory so that all subsequent output file names are to be saved in the defined output directory. Only unqualified output file names will be saved to the output directory. Fully qualified path names for output files will override this command.

Open a Wafer Map with Format

```
open "<input_file>" format <format>
```

Open an input wafer map file with the specified format. A list of valid formats can be obtained by executing the command line:

```
wmbatch64.exe -formats
```

The wafer map file will be opened, and its configuration loaded from the configuration file:

```
<config_dir>\WMFormats\<format>.txt
```

The configuration file contains critical information about the format and its framework. This information is used when parsing, converting and writing the wafer map format. It should not be necessary to edit the wafer map configuration files.

Convert Formats

```
convert format <format> [<bin_format>]
```

Convert the output database of WMLib to the specified format. Note that the WMLib input database is read only. Both databases are maintained to support certain conversion features, such as bin mapping.

For wafer map formats that support multiple bin formats, as defined in the format configuration file, the specific bin format may be specified for the conversion. This could ensure that certain bins are automatically mapped to an equivalent bin format with no loss of bin values.

Bin formats define both base and length. Examples of available bin formats are ASCII-1, HEXADECIMAL-2, DECIMAL-3, etc.

Save Wafer Map

```
save "<output_file>"
```

Save the output database to the specified file using its current wafer map format.

Select Wafer(s)

```
wafer select [<wafer_number>|all]
```

Select which wafer map is active. This applies to wafer map formats that support multiple wafer maps in one file. All subsequent commands will be applied only to the active wafer map(s) until the selection is changed. By default, all wafer maps are active.

Reload Wafer Maps

```
wafer reload
```

Reloads all the wafer maps from the input database into the output database. This is useful when there is a need to run several commands which would change the output database contents, but where each command needs to start from the same database state.

Add Null Rows/Columns

```
wafer add [top|bottom|left|right|rows|cols|all]
```

Rows or columns of null bins can be added to any side of the wafer map. Any combination of options can follow the add keyword. For example, to add a row of null bins to the bottom and a column of null bins to the left of the wafer:

```
wafer add bottom left
```

Delete All Null Rows/Columns

```
wafer delete [top|bottom|left|right|rows|cols|all]
```

All sides of the wafer specified containing only null bins will be deleted. Any combination of options can follow the delete keyword. For example, to delete all empty rows and columns around the wafer:

```
wafer delete all
```

Fill a Region of the Wafer With a Bin

```
wafer fill inside|outside <left> <top> <right> <bottom> [<bin>]
```

Change all the bins inside (or outside) the specified rectangular region to the bin provided. The coordinate system is absolute, with (0,0) in the upper left corner of the wafer. The lower right corner of the wafer has coordinates (col-1, row-1). If no bin is provided, then the null bin is used to fill the region.

Rotate the Wafer Map

```
wafer rotate <angle>
```

The wafer map will be rotated the specified angle from its current state. Since the rotation is applied to the output database only, subsequent rotations are cumulative. Rotating the wafer map may change these wafer parameters depending on the angle specified:

- Device size
- Step size
- Flat location
- Reference Device location
- Columns
- Rows
- Array

Set Device or Step Size

```
device|step size <width> <height> [<units>]
```

Set the device or step size to the width and height specified. The units may also be specified. If the units are not provided, then the size is assumed to be in the wafer units.

Set Header Tag

“<header_tag>” “<value>“

Any unrecognized keyword pair is interpreted as setting the value of a header tag. For example, this command can be used to set a common header tag, such as Wafer ID:

```
WAFER_ID 32908-03
```

[See Table A for a list of common header tags supported by WMBatch/WMLib.]

This command can also be used to set custom header tags and add arbitrary user defined information to the wafer map header:

```
MAP_AUTHOR “Artwork Conversion, Software, Inc.”
```

Note that the format of the output wafer map must be able to support this type of data.

Bin Quality

bin quality <bin> <quality>

Set the quality of the named bin. Quality values should be one of: NULL, PASS, FAIL, REFERENCE, MIRROR, EDGE, SKIP, UGLY, TEST, UNKNOWN.

Bin Description

bin description <bin> <description>

Set the description of the named bin.

Bin Format

bin format hex|dec|ascii-<length>

For wafer map formats that support multiple bin formats, this command can be used to set the current format of bins. Some conversion of bin values may change when using this command. Note that the current output wafer map format must be able to support the selected bin format. The available bin formats for each wafer map format are defined in each wafer map configuration file.

Bin Mapping

```
bin map <bin_in> <bin_out>
```

In addition to the automatic bin mapping that takes place due to a conversion command, user-defined bin mapping can also be defined. Bin values must adhere to the currently selected bin format. If they do not, they will be automatically modified.

This command will first attempt to locate the input bin value in the input wafer map and modify the value of the corresponding bin in the output wafer map (which it was originally mapped to). If the input bin value is not found in the input wafer map, the bins in the output wafer map will be examined. If a matching bin value is found, it will be mapped to the output bin value provided.

For example, if an input ASCII bin with value 1 was converted to an output decimal bin of 001, the bin could be mapped to 101 by either of these commands:

```
bin map 1 101
```

```
bin map 001 101
```

Numeric Repeat Loops

```
repeat start <n1> to <n2>
# block of commands, optionally using <#> substitution
repeat end
```

Blocks of commands can be repeated using the repeat block. This is useful for generating multiple output wafer map files from a single input, changing some parameters, and/or the output file name. The loop is controlled by the two user specified parameters n1 and n2, which are integers with n1 < n2. All the commands between the **repeat start** and **repeat end** lines will be executed once for all the values from n1 to n2. Any command inside the repeat loop can use the **<#>** substitution to use the current value of the loop. The number of # characters specifies how many digits will be used. For example, for a value of 1, <#> = 1, <###> = 001, etc. Padding using zeros will be used as needed.

Here is an example of a repeat block which would set the wafer ID to 8888-01, 8888-02, …etc. and save to an output file called WAFER-001.txt, WAFER-002.txt, … etc. for five wafer map files.

```
repeat start 1 to 5
WAFER_ID 8008-<##>
save "WAFER-<###>.txt"
repeat end
```

This repeat block would be equivalent to these commands:

```
WAFER_ID 8008-01
save "WAFER-001.txt"
WAFER_ID 8008-02
save "WAFER-002.txt"
WAFER_ID 8008-03
save "WAFER-003.txt"
WAFER_ID 8008-04
save "WAFER-004.txt"
WAFER_ID 8008-05
save "WAFER-005.txt"
```

Directory Repeat Loops

```
repeat start files in "<directory>"
# block of commands, optionally using <FILE[.EXT]> substitution
repeat end
```

The second form of the repeat block will repeat the block of commands for all files in a directory. Here is an example:

```
repeat start files in "C:\Temp\Wafer Maps\"
open <FILE.EXT> format E5-1296
convert format SECS-EG
save <FILE>.map
repeat end
```

The input file is always referred to using the substitution <FILE.EXT>. The output file can use the same substitution, or simply <FILE> and then append a different file extension.

The output files are written to the defined output directory by default, and not to the input directory, as this could cause file name collisions. To save output files to the input directory in this type of repeat block, use a full path, like this:

```
repeat start files in "C:\Temp\Wafer Maps\"
open <FILE.EXT> format E5-1296
convert format SECS-EG
save "C:\Temp\Wafer Maps\<FILE>.map"
repeat end
```

Table A – Command Header Tags

These header tags are recognized by WMBatch/WMLib to refer to common parameters defined in wafer map files. These values should always be used in command files, and not the format specific labels. The mapping between format specific and common header tags can be found in the wafer map configuration file under the HEADER section. Not all these tags can be used to set information in a wafer map, because they will be overridden based on calculation. For example, number of rows.

Note that not all formats support all common header tags. Some tags present in an input wafer map may not get converted to the output wafer map in cases where their definition is not supported.

Common Header Tag	Description
WAFER_ID	Wafer ID
LOT_ID	Lot ID
DEVICE_ID	Device or product ID
WAFER_SIZE	Wafer diameter for circular wafers
ORIGIN_LOCATION	Format specific origin location
FLAT	Flat notch location
DEVICE_X	Device with
DEVICE_Y	Device height
STEP_X	Step width
STEP_Y	Step height
NULL_BIN	Null bin value
DATE	Format specific date parameter
TIME	Format specific time parameter
ROWS	Number of rows in the array
COLUMNS	Number of columns in the array
UNITS	Units of the wafer
REF_DEVICE_COL	Reference device column
REF_DEVICE_ROW	Reference device row
BINS	A list of pass bins
SKIP_BINS	A list of skip bins
TESTER_ID	Tester ID
FRAME_ROTATION	Frame rotation
PROCESS_STATUS	Process status
WAFER_NUMBER	Wafer number

CASSETTE_NUMBER	Cassette number
SLOT_NUMBER	Slot number
TEST_NUMBER	Test number
START_TIME	Format specific time parameter
END_TIME	Format specific time parameter
OFFSET_DEVICE_X	Offset from the first device in x
OFFSET_DEVICE_Y	Offset from the first device in y
OPERATOR_NAME	Operator name
MACHINE_NUMBER	Machine Number