# 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, merging, 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.

## Import Directory

**import_dir "<dirname>"**

Define a common import directory so that all subsequent import file names are to be opened from the defined import directory. Only unqualified import file names will be opened from the import directory. Fully qualified path names for import files will override this command. Import files are used with functions that support opening multiple files simultaneously, such as merging.

## 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 a Database

**`database select input|import|output`**

Select which database is active. For most applications – opening, converting and saving a wafer map file, the output database is selected by default and need not be changed. For functionality which makes use of multiple wafer maps, such as merging, this command makes one of the three databases active. Subsequent commands would then be applied to whichever database is active. Note that only a limited number of commands can be used with the input database selected.

## 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.

## Set Wafer Coordinates

**wafer coordinates [absolute|format]**

Set which wafer coordinates to use for following commands which use position parameters. Absolute coordiantes start in the upper left corner at [0, 0] with positive columns moving right and positive rows moving down. Format coordinates use the system in use by the wafer map format, which varies between formats, but may support selection of origin, axis direction, and center offset.

## 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 by default. See the “wafer coordinates” command for information on how to change the coordinates system. If no bin is provided, then the null bin is used to fill the region.

## Set the Wafer Size

**`wafer size <x> [<y>] <units>|scale <units>`**

Set the wafer map size and units. A non-zero y value indicates a rectangular panel. Using the scale keyword with a units string will scale the existing wafer size into the new units, preserving the size.

## Set the Wafer Flat Side

**`wafer flat <bottom|left|top|right>`**

Set the wafer flat side. This changes the flat side without affecting any other wafer map parameters. It should be used to set the flat side when undefined, or to correct an invalid flat side.

## 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 the Wafer Origin

**`wafer origin <lower|upper> <left|right> | center`**

Set the wafer origin, if supported. This changes the wafer origin, which can affect the column and row values of devices in the array.

## Set Device or Step Size

**`device|step size <width> <height> [<units>]|scale <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. Using the scale keyword with a units string will scale the existing device or step size into the new units, preserving the size.

## Set a Reference Device Column and Row

**`reference device <number> <col> <row>`**

Sets the reference device number to the given column and row. The coordinate system is absolute by default. See the "wafer coordinates" command for information on how to change the coordinates system.

## Set a Reference Device XY Position

**`reference device <number> xy <x> <y> [<units>]`**

Assigns the reference device number the given xy position.

## 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**

## Bin Deletion

**bin delete unused**

Delete all unused bins from the wafer map. Unused means that the bin is defined but no device in the array references the bin. This command can be used to clean up wafer maps which have many unused bins.

## Merge Wafer Maps

**merge format <format> [<bin_format>]**

Initialize the output database to the specified wafer map format (and bin format) in preparation for a merge operation. The active wafer maps selected in the input and import databases represent the two source wafer maps to be merged. All header values from the wafer map in the input database will be copied to the output database (where supported). No header values from the import database will be copied. The number of rows and columns in the input and import wafer maps must match. The output wafer map will have the same number of rows and columns as the inputs and will be filled with NULL devices initially.

## Merge Rules

**Merge rule [NOT] [<v1>|{<v1> <v2> …}|<q1>|{<q1> <q2> …}|ANY] AND|OR [NOT] [<v1>|{<v1> <v2> …}|<q1>|{<q1> <q2> …}|ANY] OUT <bin> [<quality>]|INPUT1|INPUT2**

Define a merge rule, to be applied immediately, with the results written into the output database. Merge rules are cumulative and may change the device at a position in the array which has already been set by a previous merge rule. It is not necessary to define a merge rule mapping every device or bin. Merge rules have great flexibility, and here are some examples of their use:

```
# Set all bins that are PASS in both inputs to a PASS bin called G
merge rule PASS AND PASS OUT G PASS
# Set all bins that are FAIL in either input to a FAIL bin called X
merge rule FAIL OR FAIL OUT X FAIL
# Preserve any REFERENCE devices from the first map
merge rule REFERENCE AND ANY OUT R REFERENCE
# Application where bins 01 through 09 in the second map are defects
merge rule ANY AND {01 02 03 04 05 06 07 08 09} OUT X
# Trim devices where at least one device is NULL in the same position
merge rule NULL OR NULL OUT .
# Use the NOT keyword to group all but certain bins
merge rule NOT {01 02} AND ANY OUT F
# Use the INPUT1 or INPUT2 keywords to preserve bin values
merge rule ANY AND 01 OUT INPUT1
```

## 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 <FILE.EXT>|<FILE>.ext|file.<EXT> in
<INPUT_DIR>|<IMPORT_DIR>|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 matching files in a directory. Here is an example:

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

The input file is 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. To process all matching files in the input directory use the substitution <INPUT_DIR>:

```
repeat start <FILE.EXT> in <INPUT_DIR>
```

Only the files matching the substitution in the repeat block will be processed. For example, to only process files with the extension .abc:

```
repeat start <FILE>.abc in <INPUT_DIR>
open <FILE.EXT> format E5-1296
repeat end
```

To process all files with a file name containing xyz:

```
repeat start xyz.<EXT> in <INPUT_DIR>
open <FILE.EXT> format E5-1296
repeat end
```

In this type of repeat block, the save command can also accept a recognized header tag (see Table A) as a substitution. For example, to save all file names using the WAFER_ID as the file name:

```
repeat start <FILE.EXT> in <INPUT_DIR>
open <FILE.EXT> format E5-1296
convert format SINF
save <WAFER_ID>.sinf
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 |
| FORMAT_TYPE | Format type |
| CUSTOMER_NAME | Customer Name |