

# KiCad 命令行界面

The KiCad Team

# Table of Contents

KiCad 命令行界面简介 .....	2
封装命令 .....	3
封装导出 .....	3
封装升级 .....	3
PCB 命令 .....	5
PCB DRC .....	5
PCB 钻孔文件导出 .....	5
PCB DXF 导出 .....	6
PCB Gerber 导出：每个文件一层 .....	7
导出 PCB Gerber：每个文件可导出多个层 .....	8
PCB GLB 导出 .....	9
PCB IPC-2581 导出 .....	10
PCB PDF 导出 .....	11
PCB 位置文件导出 .....	12
PCB STEP 导出 .....	13
PCB SVG 导出 .....	14
PCB VRML 导出 .....	15
原理图命令 .....	17
原理图 ERC .....	17
原理图物料清单导出 .....	17
原理图 DXF 导出 .....	19
原理图 HPGL 导出 .....	20
原理图网表导出 .....	21
原理图 PDF 导出 .....	21
原理图 PostScript 导出 .....	22
原理图物料清单导出（旧版 BOM 脚本） .....	23
原理图 SVG 导出 .....	23
符号命令 .....	25
符号导出 .....	25
符号升级 .....	25
版本命令 .....	27

## 参考手册

### Copyright

本文档的版权归（C）2023-2024 所有，由下列贡献者提供。您可以根据 GNU 通用公共许可证（<http://www.gnu.org/licenses/gpl.html>）版本 3 或更高版本或知识共享署名许可证（<http://creativecommons.org/licenses/by/3.0/>）版本 3.0 或更高版本的条款分发和/或修改它。

本指南中的所有商标均属于其合法所有者。

### Contributors

Graham Keeth

### 翻译人员

taotieren <[admin@taotieren.com](mailto:admin@taotieren.com)>, 2019-2024.

### Feedback

KiCad 项目欢迎与软件或其文档相关的反馈、错误报告和建议。有关如何提交反馈或报告问题的更多信息，请参阅 <https://www.kicad.org/help/report-an-issue/> 上的说明

# KiCad 命令行界面简介

KiCad 提供了一个命令行界面，可以通过运行 `kicad-cli` 二进制文件来使用。使用命令行界面，您可以以自动化方式对原理图、PCB、符号和封装执行许多操作，例如绘制 PCB 设计中的 Gerber 文件或将符号库从传统文件格式升级到现代格式。

## NOTE

在 macOS 上，`kicad-cli` 可执行文件位于 `/Applications/KiCad/KiCad.app/Contents/MacOS/kicad-cli`。

`kicad-cli` 命令有 5 个子命令：`fp`、`pcb`、`sch`、`sym` 和 `version`。每个子命令都可以有自己的子命令和参数。例如，要从 PCB 导出 Gerber 文件，您可以运行 `kicad-cli pcb export gerbers example.kicad_pcb`。

您可以添加 `--help` 或 `-h` 标志来查看有关每个子命令的信息。例如，运行 `kicad-cli pcb -h` 打印有关 `pcb` 子命令的使用信息，而 `kicad-cli pcb export gerbers -h` 专门打印 `pcb export gerbers` 子命令的使用信息。

# 封装命令

`fp` 子命令将封装导出为另一种格式，或将封装库升级到 KiCad 封装文件格式的当前版本。

## 封装导出

`fp export svg` 命令将指定库中的一个或多个封装导出到 SVG 文件中。

Usage: `kicad-cli fp export svg [--help] [--output OUTPUT_FILE] [--layers LAYER_LIST] [--define-var KEY=VALUE] [--theme VAR] [--footprint FOOTPRINT_NAME] [--black-and-white] INPUT_DIR`

位置参数：

INPUT_DIR	要导出的封装库目录（.pretty）。
-----------	---------------------

可选参数：

<code>-h, --help</code>	显示封装 SVG 导出命令的帮助。
<code>-o &lt;output dir&gt;, --output &lt;output dir&gt;</code>	导出文件的输出目录。不使用该参数时，文件将导出到当前目录。
<code>-l &lt;layer list&gt;, --layers &lt;layer list&gt;</code>	以逗号分隔的要从封装中导出的图层名称列表，如 F.Cu,B.Cu。如果没有给出图层，则导出所有图层。
<code>-D &lt;variable name&gt;=&lt;value&gt;, --define-var &lt;variable_name&gt;=&lt;value&gt;</code>	添加或覆盖工程变量定义。可多次使用以定义多个变量。
<code>-t &lt;theme name&gt;, --theme &lt;theme name&gt;</code>	用于导出的主题名称。如果没有给出主题，则使用封装编辑器当前选择的主题。
<code>--fp &lt;footprint&gt;, --footprint &lt;footprint&gt;</code>	要从库中导出的特定封装的名称。不使用此参数时，将导出库中的所有封装。
<code>--black-and-white</code>	导出黑白封装。

## 封装升级

`fp upgrade` 命令将指定的封装库从传统封装格式升级到当前版本的 KiCad 的原生格式。如果输入库已采用当前文件格式，则不执行任何操作。

用法：`kicad-cli fp upgrade [-h] [--output OUTPUT_DIR] [--force] INPUT_DIR`

位置参数：

INPUT_DIR	要升级的封装库目录（.pretty）。
-----------	---------------------

可选参数：

<code>-h, --help</code>	显示封装升级命令的帮助。
<code>-o &lt;output dir&gt;, --output &lt;output dir&gt;</code>	升级后封装的输出目录。不使用此参数时，升级后的封装会保存在原始封装之上。
<code>--force</code>	强制保存输入库，即使它已经是当前文件格式。

# PCB 命令

`pcb` 命令运行设计规则检查或将电路板导出为各种其他文件格式，包括制造和 3D 文件。

## PCB DRC

`pcb drc` 命令在电路板上运行设计规则检查并生成报告。

```
Usage: kicad-cli pcb drc [--help] [--output OUTPUT_FILE] [--define-var KEY=VALUE] [--format
FORMAT] [--all-track-errors] [--schematic-parity] [--units UNITS] [--severity-all] [--severity-
error] [--severity-warning] [--severity-exclusions] [--exit-code-violations] INPUT_FILE
```

位置参数：

<code>INPUT_FILE</code>	要在其上运行 DRC 的电路板文件。
<code>-h, --help</code>	显示 DRC 命令的帮助。
<code>-o &lt;output filename&gt;, --output &lt;output filename&gt;</code>	生成的 DRC 报告的输出文件名。不使用该参数时，输出文件名将与输入文件相同，根据所选格式，文件扩展名为 <code>.rpt</code> 或 <code>.json</code> 。
<code>-D &lt;variable name&gt;=&lt;value&gt;, --define-var &lt;variable_name&gt;=&lt;value&gt;</code>	添加或覆盖工程变量定义。可多次使用以定义多个变量。
<code>--format &lt;format&gt;</code>	报告文件格式。选项为 <code>report</code> （默认）或 <code>json</code> 。
<code>--all-track-errors</code>	报告每个布线的所有错误。
<code>--schematic-parity</code>	测试 PCB 和原理图之间的奇偶校验。
<code>--units &lt;unit&gt;</code>	要在报表中使用的单位。选项为 <code>mm</code> （默认）、 <code>in</code> 或 <code>mils</code> 。
<code>--severity-all</code>	报告所有 DRC 违规行为。这相当于使用所有其他 DRC 严重性选项。
<code>--severity-error</code>	报告所有错误级别的 DRC 违规行为。该选项可与其他 DRC 严重性选项结合使用。
<code>--severity-warning</code>	报告所有警告级别的 DRC 违规行为。该选项可与其他 DRC 严重性选项结合使用。
<code>--severity-exclusions</code>	报告所有排除在外的 DRC 违规行为。该选项可与其他 DRC 严重性选项结合使用。
<code>--exit-code-violations</code>	根据是否存在 DRC 违规，返回退出代码。如果没有发现违规，则退出代码为 0；如果发现违规，则退出代码为 5。

## PCB 钻孔文件导出

`pcb export drill` 命令从电路板导出钻孔文件。

Usage: kicad-cli pcb export drill [--help] [--output OUTPUT\_DIR] [--format FORMAT] [--drill-origin DRILL\_ORIGIN] [--excellon-zeros-format ZEROS\_FORMAT] [--excellon-oval-format OVAL\_FORMAT] [--excellon-units UNITS] [--excellon-mirror-y] [--excellon-min-header] [--excellon-separate-th] [--generate-map] [--map-format MAP\_FORMAT] [--gerber-precision VAR] INPUT\_FILE

位置参数：

INPUT_FILE	要导出的电路板文件。
------------	------------

可选参数：

-h, --help	显示钻孔文件导出命令的帮助。
-o <output dir>, --output <output dir>	钻孔文件的输出目录。不使用此参数时，钻孔文件保存在当前目录下。
--format <format>	钻取文件格式。选项为 excellon（默认）或 gerber。
--drill-origin <origin>	钻孔文件的坐标原点。选项有 absolute（默认），用于使用电路板的绝对原点；或 plot，用于使用电路板的钻孔/置位原点。
--excellon-zeros-format <format>	钻孔文件中零的表示格式。提供四种选项：decimal（默认，采用十进制格式）、suppressleading（省略前导零）、suppresstrailing（省略尾随零）、以及 keep（保留所有零）。这些设置仅在 Excellon 格式的钻孔文件中有效。
--excellon-oval-format <format>	控制椭圆形通孔的钻孔策略。提供两种模式：route（布线模式）和 alternate（备用模式，默认选项）。这些设置仅在 Excellon 格式的钻孔文件中生效。
-U <units>, --excellon-units <units>	钻孔文件的单位。选项为 mm（默认）或 in。仅适用于 Excellon 格式的钻孔文件。
--excellon-mirror-y	沿 Y 方向镜像钻孔文件。仅适用于 Excellon 格式的钻孔文件。
--excellon-min-header	在钻孔文件中使用最小的钻头。仅适用于 Excellon 格式的钻孔文件。
--excellon-separate-th	为电镀通孔和非电镀通孔生成不同的钻孔文件。仅适用于 Excellon 格式的钻孔文件。
--generate-map	除了钻取文件之外，还生成映射文件。
--map-format <format>	映射文件格式。选项包括 pdf（默认）、gerberx2、ps、dxf 或 svg。
--gerber-precision <precision>	钻孔文件的精度（位数）。有效选项为 5 或 6（默认）。仅适用于 Gerber 格式的钻孔文件。

## PCB DXF 导出

pcb export dxf 命令将电路板设计导出为 DXF 文件。

Usage: kicad-cli pcb export dxf [--help] [--output OUTPUT\_FILE] [--layers LAYER\_LIST] [--drawing-sheet SHEET\_PATH] [--define-var KEY=VALUE] [--exclude-refdes] [--exclude-value] [--use-contours]



`[--include-border-title] [--output-units UNITS] INPUT_FILE`

位置参数：

INPUT_FILE	要导出的电路板文件。
------------	------------

可选参数：

<code>-h, --help</code>	显示 DXF 导出命令的帮助。
<code>-o &lt;output filename&gt;, --output &lt;output filename&gt;</code>	输出文件名。如果不使用此参数，输出文件名将与输入文件相同，文件扩展名为 .dxf。
<code>--drawing-sheet &lt;sheet path&gt;</code>	在绘制中使用的图纸的路径，覆盖电路板文件中指定的图纸。
<code>-D &lt;variable name&gt;=&lt;value&gt;, --define-var &lt;variable_name&gt;=&lt;value&gt;</code>	添加或覆盖工程变量定义。可多次使用以定义多个变量。
<code>-l &lt;layer list&gt;, --layers &lt;layer list&gt;</code>	以逗号分隔的要从封装中导出的图层名称列表，如 F.Cu,B.Cu。必须至少给出一个图层。
<code>--erd, --exclude-refdes</code>	从绘制中排除封装位号。
<code>--ev, --exclude-value</code>	从绘制中排除封装值。
<code>--uc, --use-contours</code>	使用轮廓线绘制图形项目。
<code>-ibt, --include-border-title</code>	在绘制中包括图纸边框和标题块。
<code>--ou &lt;unit&gt;, --output-units &lt;unit&gt;</code>	输出单位。选项为 mm 或 in（默认）。

## PCB Gerber 导出：每个文件一层

`pcb export gerbers` 命令将电路板设计导出到 Gerber 文件，每个文件一层。

### NOTE

请注意有两种不同的 Gerber 导出命令，即 `gerber` 和 `gerbers`。`gerber` 命令将多个 PCB 层绘制到单个 Gerber 文件中，而 `gerbers` 命令将多个 Gerber 文件绘制到单个 Gerber 文件中。在制作 PCB 时，通常应使用 `gerbers` 命令。

Usage: `kicad-cli pcb export gerbers [--help] [--output OUTPUT_FILE] [--layers LAYER_LIST] [--drawing-sheet SHEET_PATH] [--define-var KEY=VALUE] [--exclude-refdes] [--exclude-value] [--include-border-title] [--no-x2] [--no-netlist] [--subtract-soldermask] [--disable-aperture-macros] [--use-drill-file-origin] [--precision PRECISION] [--no-protel-ext] [--common-layers COMMON_LAYER_LIST] [--board-plot-params] INPUT_FILE`

位置参数：

INPUT_FILE	要导出的电路板文件。
------------	------------

可选参数：

<code>-h, --help</code>	显示 Gerber 导出命令的帮助。
<code>-o &lt;output dir&gt;, --output &lt;output dir&gt;</code>	导出文件的输出文件夹。不使用该参数时，文件将导出到当前目录。
<code>-l &lt;layer list&gt;, --layers &lt;layer list&gt;</code>	以逗号分隔的图层名称列表，如 F.Cu,B.Cu。如果未使用此参数，则将绘制所有图层。
<code>--drawing-sheet &lt;sheet path&gt;</code>	在绘制中使用的图纸的路径，覆盖电路板文件中指定的图纸。
<code>-D &lt;variable name&gt;=&lt;value&gt;, --define-var &lt;variable_name&gt;=&lt;value&gt;</code>	添加或覆盖工程变量定义。可多次使用以定义多个变量。
<code>--erd, --exclude-refdes</code>	从绘制中排除封装位号。
<code>--ev, --exclude-value</code>	从绘制中排除封装值。
<code>--ibt, --include-border-title</code>	包括页面边框和标题块。
<code>--no-x2</code>	请勿使用扩展的 X2 格式。
<code>--no-netlist</code>	不要包含网表属性。
<code>--subtract-soldermask</code>	去除没有阻焊层区域的丝印。
<code>--disable-aperture-macros</code>	禁用光圈宏。
<code>--use-drill-file-origin</code>	使用钻孔/放置文件原点，而不是绝对原点。
<code>--precision &lt;precision&gt;</code>	Gerber 文件的精度（位数）。有效选项为 5 或 6（默认）。
<code>--no-protel-ext</code>	使用 .gbr 文件扩展名而不是 Protel 文件扩展名（.gbl、.gtl 等）。
<code>--cl &lt;layer list&gt;, --common-layers &lt;layer list&gt;</code>	以逗号分隔的图层名称列表，用于在所有图层上绘制，如 F.Cu,B.Cu。
<code>--board-plot-params</code>	使用已在电路板文件中配置的 Gerber 绘制设置。

## 导出 PCB Gerber：每个文件可导出多个层

`pcb export gerber` 命令将一个或多个板层导出到单个 Gerber 文件。

### NOTE

请注意有两种不同的 Gerber 导出命令，即 `gerber` 和 `gerbers`。`gerber` 命令将多个 PCB 层绘制到单个 Gerber 文件中，而 `gerbers` 命令将多个 Gerber 文件绘制到单个 Gerber 文件中。在制作 PCB 时，通常应使用 `gerbers` 命令。

Usage: kicad-cli pcb export gerber [--help] [--output OUTPUT\_FILE] [--layers LAYER\_LIST] [--drawing-sheet SHEET\_PATH] [--define-var KEY=VALUE] [--exclude-refdes] [--exclude-value] [--include-border-title] [--no-x2] [--no-netlist] [--subtract-soldermask] [--disable-aperture-macros] [--use-drill-file-origin] [--precision PRECISION] [--no-protel-ext] INPUT\_FILE

位置参数：

INPUT_FILE	要导出的电路板文件。
------------	------------

可选参数：

-h, --help	显示 Gerber 导出命令的帮助。
-o <output filename>, --output <output filename>	输出文件名。不使用此参数时，输出文件名将与输入文件相同，文件扩展名为 .gbr。
-l <layer list>, --layers <layer list>	以逗号分隔的要从电路板上绘制的图层名称列表，如 F.Cu,B.Cu。所有层都将绘制到输出文件中。必须至少给出一个层。
--drawing-sheet <sheet path>	绘制时使用的绘制表路径，覆盖电路板文件中指定的绘制表。优先于电路板文件中指定的绘制纸。
-D <variable name>=<value>, --define-var <variable_name>=<value>	添加或覆盖工程变量定义。可多次使用以定义多个变量。
--erd, --exclude-refdes	从绘制中排除封装位号。
--ev, --exclude-value	从绘制中排除封装值。
--ibt, --include-border-title	包含工作表边框和标题块。
--no-x2	不使用扩展 X2 格式。
--no-netlist	不包含网表属性。
--subtract-soldermask	去除无阻焊区域的丝印。
--disable-aperture-macros	禁用光圈宏。
--use-drill-file-origin	使用钻孔/放置文件原点，而不是绝对原点。
--precision <precision>	Gerber 文件的精度（位数）。有效选项为 5 或 6（默认）。
--no-protel-ext	使用 .gbr 文件扩展名代替 Protel 文件扩展名（.gbl、.gtl 等）。

## PCB GLB 导出

pcb export glb 命令将电路板设计导出为 GLB（二进制 GLTF）3D 文件。

Usage: kicad-cli pcb export glb [--help] [--output OUTPUT\_FILE] [--define-var KEY=VALUE] [--force] [--grid-origin] [--drill-origin] [--no-unspecified] [--no-dnp] [--subst-models] [--board-

only] [--include-tracks] [--include-zones] [--min-distance MIN\_DIST] [--user-origin VAR] INPUT\_FILE

位置参数：

INPUT_FILE	要导出的电路板文件。
------------	------------

可选参数：

-h, --help	显示 GLB 导出命令的帮助。
-o <output filename>, --output <output filename>	输出文件名。不使用该参数时，输出文件名将与输入文件相同，文件扩展名为 .glb。
-D <variable name>=<value>, --define-var <variable_name>=<value>	添加或覆盖工程变量定义。可多次使用以定义多个变量。
-f, --force	强制覆盖输出文件。
--grid-origin	使用网格原点作为输出文件的原点。
--drill-origin	使用钻孔原点作为输出文件的原点。
--no-unspecified	排除封装类型为 "未指定" 的部件的 3D 模型。
--no-dnp	排除带有 "不填充" 属性的 3D 模型。
--subst-models	用同名 STEP 模型替换封装中的 VRML 模型（如果存在）。
--board-only	在生成的模型中只包含电路板本身；不包含所有元件模型。
--include-tracks	在导出时包含布线（耗时）。
--include-zones	导出时包含区域（耗时）。
--min-distance <min distance>	认为两个点在同一位置的误差。默认值：0.01mm。
--user-origin <output origin>	为输出文件指定自定义的原点，包括 X 和 Y 坐标。例如，1x1in、1x1inch 或 25.4x25.4mm。默认单位是毫米。

## PCB IPC-2581 导出

pcb export ipc2581 命令以 IPC-2581 格式导出电路板设计。

Usage: kicad-cli pcb export ipc2581 [--help] [--output OUTPUT\_FILE] [--drawing-sheet SHEET\_PATH] [--define-var KEY=VALUE] [--precision PRECISION] [--compress] [--version VAR] [--units VAR] [--bom-col-int-id FIELD\_NAME] [--bom-col-mfg-pn FIELD\_NAME] [--bom-col-mfg FIELD\_NAME] [--bom-col-dist-pn FIELD\_NAME] [--bom-col-dist FIELD\_NAME] INPUT\_FILE

INPUT_FILE	要导出的电路板文件。
------------	------------

可选参数：

<code>-h, --help</code>	显示 IPC-2581 输出命令的帮助。
<code>-o &lt;output filename&gt;, --output &lt;output filename&gt;</code>	输出文件名。不使用此参数时，输出文件名将与输入文件相同，文件扩展名为 .xml。
<code>--drawing-sheet &lt;sheet path&gt;</code>	在绘图中使用的图纸路径，优先于电路板文件中指定的图纸。
<code>-D &lt;variable name&gt;=&lt;value&gt;, --define-var &lt;variable_name&gt;=&lt;value&gt;</code>	添加或覆盖工程变量定义。可多次使用以定义多个变量。
<code>--precision &lt;precision&gt;</code>	导出文件的精度（小数点后的位数）。默认为 3。
<code>--compress</code>	将输出文件压缩为 ZIP 文件。
<code>--version &lt;IPC-2581 standard version&gt;</code>	要使用的 IPC-2581 标准版本。选项为 B 或 C（默认）。
<code>--units</code>	导出时要使用的单位。选项为 mm（默认）或 in。
<code>--bom-col-int-id</code>	定义 BOM 中零件内部 ID 的列标题。可以是任何封装字段，或留空以省略该列。
<code>--bom-col-mfg-pn</code>	定义 BOM 中制造商零件编号名称的列标题。可以是任何封装字段，或留空以省略该列。
<code>--bom-col-mfg</code>	定义 BOM 中制造商零件的列标题。可以是任何封装字段，或留空以省略该列。
<code>--bom-col-dist-pn</code>	定义 BOM 中分销商零件编号名称的列标题。可以是任何封装字段，或留空以省略该列。
<code>--bom-col-dist</code>	定义 BOM 中分销商零件的列标题。可以是任何封装字段，或留空以省略该列。

## PCB PDF 导出

将 `pcb export pdf` 命令导出电路板设计为 PDF 文件。

Usage: `kicad-cli pcb export pdf` [`--help`] [`--output OUTPUT_FILE`] [`--layers LAYER_LIST`] [`--drawing-sheet SHEET_PATH`] [`--define-var KEY=VALUE`] [`--mirror`] [`--exclude-refdes`] [`--exclude-value`] [`--include-border-title`] [`--negative`] [`--black-and-white`] [`--theme THEME_NAME`] [`--drill-shape-opt VAR`] `INPUT_FILE`

位置参数：

<code>INPUT_FILE</code>	要导出的电路板文件。
-------------------------	------------

可选参数：

<code>-h, --help</code>	显示 PDF 导出命令的帮助信息。
<code>-o &lt;output filename&gt;, --output &lt;output filename&gt;</code>	指定输出文件名。如果不使用此参数，输出文件名将与输入文件相同，但后缀为 .pdf。
<code>-l &lt;layer list&gt;, --layers &lt;layer list&gt;</code>	从电路板导出的图层名称列表，以逗号分隔，如 F.Cu,B.Cu。至少需要指定一个图层。
<code>--drawing-sheet &lt;sheet path&gt;</code>	指定在绘制过程中使用的图纸路径，覆盖电路板文件中指定的图纸。
<code>-D &lt;variable name&gt;=&lt;value&gt;, --define-var &lt;variable_name&gt;=&lt;value&gt;</code>	添加或覆盖工程变量定义。可多次使用以定义多个变量。
<code>-m, --mirror</code>	镜像电路板。这对于显示底层可能很有用。
<code>--erd, --exclude-refdes</code>	在绘制时排除元件封装的位号。
<code>--ev, --exclude-value</code>	在绘制时排除元件封装的值。
<code>--ibt, --include-border-title</code>	包含图纸边框和标题栏。
<code>-n, --negative</code>	以负片形式绘制。
<code>--black-and-white</code>	以黑白模式绘制。
<code>-t &lt;theme name&gt;, --theme &lt;theme name&gt;</code>	指定导出时使用主题的名称。如果没有给出主题，将使用电路板编辑器当前选定的主题。
<code>--drill-shape-opt</code>	绘制中钻孔标记的形状选项。可选值为 0（无钻孔标记）、1（小标记）或 2（实际尺寸标记，默认）。

## PCB 位置文件导出

`pcb export pos` 命令从电路板设计中导出位置文件。

```
Usage: kicad-cli pcb export pos [--help] [--output OUTPUT_FILE] [--side VAR] [--format FORMAT] [--units UNITS] [--bottom-negate-x] [--use-drill-file-origin] [--smd-only] [--exclude-fp-th] [--exclude-dnp] [--gerber-board-edge] INPUT_FILE
```

位置参数：

<code>INPUT_FILE</code>	要导出的电路板文件。
-------------------------	------------

可选参数：

<code>-h, --help</code>	显示位置文件导出命令的帮助信息。
<code>-o &lt;output filename&gt;, --output &lt;output filename&gt;</code>	设置输出文件名。如果不使用此参数，输出文件名将与输入文件相同，但后缀为 <code>.pos</code> 。
<code>--side &lt;side&gt;</code>	指定要导出的电路板面。选项有 <code>front</code> （正面）、 <code>back</code> （背面）或 <code>both</code> （双面，默认）。Gerber 格式不支持 <code>both</code> 选项。
<code>--format &lt;format&gt;</code>	位置文件的格式。选项有 <code>ascii</code> （默认）、 <code>csv</code> 或 <code>gerber</code> 。
<code>--units &lt;unit&gt;</code>	位置文件中的单位。选项有 <code>in</code> （英寸，默认）或 <code>mm</code> （毫米）。此选项对 Gerber 格式无效。
<code>--bottom-negate-x</code>	对底层的元件封装使用负 X 坐标。此选项对 Gerber 格式无效。
<code>--use-drill-file-origin</code>	使用钻孔/放置文件的原点而非绝对原点。此选项对 Gerber 格式无效。
<code>--smd-only</code>	只包含表面贴装技术 (SMT) 元件。此选项对 Gerber 格式无效。
<code>--exclude-fp-th</code>	排除所有带有通孔焊盘的封装。此选项对 Gerber 格式无效。
<code>--exclude-dnp</code>	排除所有具有“不安装” (DNP) 属性的封装。
<code>--gerber-board-edge</code>	导出时包含电路板边缘层（仅限 Gerber 格式）。

## PCB STEP 导出

`pcb export step` 命令将电路板设计导出为 STEP 文件。

```
Usage: kicad-cli pcb export step [--help] [--output OUTPUT_FILE] [--define-var KEY=VALUE] [--force] [--grid-origin] [--drill-origin] [--no-unspecified] [--no-dnp] [--subst-models] [--board-only] [--include-tracks] [--include-zones] [--min-distance MIN_DIST] [--no-optimize-step] [--user-origin VAR] INPUT_FILE
```

位置参数：

<code>INPUT_FILE</code>	要导出的电路板文件。
-------------------------	------------

可选参数：

<code>-h, --help</code>	显示 STEP 文件导出命令的帮助信息。
<code>-o &lt;output filename&gt;, --output &lt;output filename&gt;</code>	设置输出文件名。若省略此参数，输出文件名会与输入文件相同，仅更改后缀为 .step。
<code>-D &lt;variable name&gt;=&lt;value&gt;, --define-var &lt;variable_name&gt;=&lt;value&gt;</code>	添加或重置工程的变量定义。可以多次使用以定义多个变量。
<code>-f, --force</code>	强制覆盖输出文件。
<code>--grid-origin</code>	使用网格原点作为输出文件的坐标原点。
<code>--drill-origin</code>	使用钻孔原点作为输出文件的坐标原点。
<code>--no-unspecified</code>	排除封装类型为“未指定”的元件 3D 模型。
<code>--no-dnp</code>	排除具有“不安装” (DNP) 属性的元件 3D 模型。
<code>--subst-models</code>	若有同名的 STEP 模型，则替换封装中的 VRML 模型。
<code>--board-only</code>	仅包含电路板自身在生成的模型中；排除所有元件模型。
<code>--include-tracks</code>	导出时包括线路（操作耗时）。
<code>--include-zones</code>	导出时包括区域（操作耗时）。
<code>--min-distance &lt;min distance&gt;</code>	定义判断两点位于同一位置的误差。默认值：0.01 mm。
<code>--no-optimize-step</code>	不对 STEP 文件进行优化。这样可以写入参数化曲线，减小文件大小和读写时间，但可能会影响与其他软件的兼容性。
<code>--user-origin &lt;output origin&gt;</code>	自定义输出文件的坐标原点，包括 X 和 Y 坐标。例如，1x1in、1x1inch 或 25.4x25.4mm。默认单位是毫米。

## PCB SVG 导出

`pcb export svg` 命令将电路板设计导出为 SVG 文件。

Usage: `kicad-cli pcb export svg` [`--help`] [`--output OUTPUT_FILE`] [`--layers LAYER_LIST`] [`--drawing-sheet SHEET_PATH`] [`--define-var KEY=VALUE`] [`--mirror`] [`--theme THEME_NAME`] [`--negative`] [`--black-and-white`] [`--page-size-mode MODE`] [`--exclude-drawing-sheet`] [`--drill-shape-opt SHAPE_OPTION`] `INPUT_FILE`

位置参数：

<code>INPUT_FILE</code>	要导出的电路板文件。
-------------------------	------------

可选参数：



<code>-h, --help</code>	显示 SVG 文件导出命令的帮助信息。
<code>-o &lt;output filename&gt;, --output &lt;output filename&gt;</code>	设定输出文件名。如果不使用此选项，输出文件名将与输入文件相同，仅修改扩展名为 <code>.svg</code> 。
<code>-l &lt;layer list&gt;, --layers &lt;layer list&gt;</code>	一个由逗号分隔的图层名称列表，用于从电路板导出，例如 <code>F.Cu,B.Cu</code> 。至少需要给出一个图层。
<code>--drawing-sheet &lt;sheet path&gt;</code>	指定用于绘图的图纸路径，覆盖电路板文件中指定的图纸。
<code>-D &lt;variable name&gt;=&lt;value&gt;, --define-var &lt;variable_name&gt;=&lt;value&gt;</code>	添加或重设工程变量定义。可多次使用以定义多个变量。
<code>-m, --mirror</code>	镜像电路板。这对于显示底层很有帮助。
<code>-t &lt;theme name&gt;, --theme &lt;theme name&gt;</code>	指定用于导出的主题名称。如果未给出主题，将使用电路板编辑器当前选定的主题。
<code>-n, --negative</code>	以负片模式绘制。
<code>--black-and-white</code>	以黑白模式绘制。
<code>--page-size-mode &lt;mode&gt;</code>	设置页面尺寸模式。选项为 0（默认），1，或 2。0 使输出页面尺寸适应整个图纸，包括图纸框架和标题栏。1 使输出页面尺寸匹配当前页面尺寸。2 使输出页面尺寸与电路板本身尺寸一致。
<code>--exclude-drawing-sheet</code>	绘制 SVG 时不包含图纸。
<code>--drill-shape-opt</code>	设定绘图中钻孔标记的形状。选项为 0 无钻孔标记，1 小标记，或 2 实际尺寸标记（默认）。

## PCB VRML 导出

`pcb export vrml` 命令将电路板设计导出为 VRML 3D 文件。

Usage: `kicad-cli pcb export vrml [--help] [--output OUTPUT_FILE] [--define-var KEY=VALUE] [--force] [--user-origin VAR] [--units VAR] [--models-dir VAR] [--models-relative] INPUT_FILE`

位置参数：

<code>INPUT_FILE</code>	要导出的电路板文件。
-------------------------	------------

可选参数：

<code>-h, --help</code>	显示 VRML 导出命令的帮助信息。
<code>-o &lt;output filename&gt;, --output &lt;output filename&gt;</code>	输出文件名。当没有使用这个参数时，输出文件名将与输入文件相同，只是扩展名改为 .wrl。
<code>-D &lt;variable name&gt;=&lt;value&gt;, --define-var &lt;variable_name&gt;=&lt;value&gt;</code>	添加或覆盖工程变量定义。可以多次使用来定义多个变量。
<code>-f, --force</code>	强制覆盖输出文件。
<code>--user-origin &lt;output origin&gt;</code>	指定输出文件的自定义原点，带有 X 和 Y 坐标。例如，1x1in, 1x1inch, 或 25.4x25.4mm。默认单位是毫米。如果没有给出这个选项，将使用板中心作为原点。
<code>--units &lt;units&gt;</code>	输出文件中使用的单位。选项有 mm, m, in (默认), 或 tenths (英寸的十分之一)。
<code>--models-dir &lt;output model directory&gt;</code>	将元件模型复制到的输出目录名称。如果不使用，元件模型将嵌入到输出文件中。
<code>--models-relative</code>	在使用 --models-dir 时，在输出文件中使用相对路径。

# 原理图命令

`sch` 命令用于运行电气规则检查，将原理图导出为其他各种文件格式，或导出物料清单或网表。每个子命令都有自己的选项。

## 原理图 ERC

`sch erc` 命令对原理图运行电气规则检查并生成报告。

Usage: `kicad-cli sch erc` [--help] [--output OUTPUT\_FILE] [--define-var KEY=VALUE] [--format VAR] [--units VAR] [--severity-all] [--severity-error] [--severity-warning] [--severity-exclusions] [--exit-code-violations] INPUT\_FILE

位置参数：

INPUT_FILE	运行 ERC 的原理图文件。
------------	----------------

<code>-h, --help</code>	显示 ERC 命令的帮助信息。
<code>-o &lt;output filename&gt;, `--output &lt;output filename&gt;</code>	生成的 ERC 报告的输出文件名。如果不使用此参数，输出文件名将与输入文件相同，后缀为 <code>.rpt</code> 或 <code>.json</code> ，具体取决于所选的格式。
<code>-D &lt;variable name&gt;=&lt;value&gt;, --define-var &lt;variable_name&gt;=&lt;value&gt;</code>	添加或覆盖工程变量定义。可以多次使用以定义多个变量。
<code>--format &lt;format&gt;</code>	报告文件格式。选项有 <code>report</code> （默认）或 <code>json</code> 。
<code>--units &lt;unit&gt;</code>	报告中使用的单位。选项有 <code>mm</code> （默认）、 <code>in</code> 或 <code>mils</code> 。
<code>--severity-all</code>	报告所有 ERC 违规。这相当于使用所有其他 ERC 严重性选项。
<code>--severity-error</code>	报告所有错误级别的 ERC 违规。可以与其他 ERC 严重性选项组合使用。
<code>--severity-warning</code>	报告所有警告级别的 ERC 违规。可以与其他 ERC 严重性选项组合使用。
<code>--severity-exclusions</code>	报告所有被排除的 ERC 违规。可以与其他 ERC 严重性选项组合使用。
<code>--exit-code-violations</code>	根据是否存在 ERC 违规返回退出代码。如果未发现违规，则退出代码为 0；如果找到任何违规，则退出代码为 5。

## 原理图物料清单导出

`sch export bom` 命令从原理图中导出 BOM。BOM 导出具有许多用于控制格式和包含字段的选项。此导出方法等同于 [导出 BOM](#) 从符号字段表。

### NOTE

要使用旧版 XML 和 Python BOM 脚本工作流导出 BOM，请使用 `sch export python-bom` 命令。

Usage: `kicad-cli sch export bom` [--help] [--output OUTPUT\_FILE] [--preset PRESET] [--format-preset FMT\_PRESET] [--fields FIELDS] [--labels LABELS] [--group-by GROUP\_BY] [--sort-field

```
SORT_BY] [--sort-asc] [--filter FILTER] [--exclude-dnp] [--field-delimiter FIELD_DELIM] [--string-delimiter STR_DELIM] [--ref-delimiter REF_DELIM] [--ref-range-delimiter REF_RANGE_DELIM] [--keep-tabs] [--keep-line-breaks] INPUT_FILE
```

位置参数：

INPUT_FILE	要导出的原理图文件。
------------	------------

可选参数：

<code>-h, --help</code>	显示帮助信息并退出。
<code>-o &lt;output filename&gt;, --output &lt;output filename&gt;</code>	输出文件的名称。如果不使用此参数，输出文件名将与输入文件相同，但具有 <code>.csv</code> 的文件扩展名。
<code>--preset &lt;preset&gt;</code>	使用原理图中的命名 BOM 预设设置，例如 "按值分组"。
<code>--format-preset &lt;format preset&gt;</code>	使用原理图中的命名 BOM 格式预设设置，例如 CSV。
<code>--fields &lt;fields&gt;</code>	要导出的有序字段列表。* 表示包含所有字段。特殊符号字段如不安装 (DNP) 或排除在电路板上 (Exclude from board) 可以通过 <code>\${DNP}</code> 或 <code>\${EXCLUDE_FROM_BOARD}</code> 访问 (参见文本变量文档以获取字段列表)。默认: "Reference (位号), Value (值), Footprint (封装), \${QUANTITY} (数量), \${DNP} (不安装)"。
<code>--labels &lt;labels&gt;</code>	应用于导出字段的有序标签列表 (默认: "Refs (位号), Value (值), Footprint (封装), Qty (数量), DNP (不安装)")。
<code>--group-by &lt;fields&gt;</code>	当字段值匹配时，用于对位号进行分组的字段。
<code>--sort-field &lt;fields&gt;</code>	排序依据的字段名称 (默认: "Reference (位号)")。
<code>--sort-asc</code>	如果给出，则按升序排序。如果没有给出，则按降序排序。
<code>--filter &lt;filter&gt;</code>	移除输出行的过滤字符串。
<code>--exclude-dnp</code>	排除具有“不安装” (DNP) 属性的符号。
<code>--field-delimiter &lt;delimiter&gt;</code>	输出字段/列之间的分隔符 (默认: ";")。
<code>--string-delimiter &lt;delimiter&gt;</code>	包裹字段的字符 (默认不使用)。
<code>--ref-delimiter &lt;delimiter&gt;</code>	单个位号之间的分隔符 (默认: ";")。
<code>--ref-range-delimiter &lt;delimiter&gt;</code>	位号范围内的分隔符 (默认: "-")。留空则无范围。
<code>--keep-tabs</code>	保留输入字段中的制表符。默认会去除。
<code>--keep-line-breaks</code>	保留输入字段中的换行符。默认会去除。

## 原理图 DXF 导出

`sch export dxf` 命令将原理图导出到 DXF 文件。设计中的每个图纸都导出到其自己的文件中。

```
Usage: kicad-cli sch export dxf [--help] [--output OUTPUT_DIR] [--drawing-sheet SHEET_PATH] [--define-var KEY=VALUE] [--theme THEME_NAME] [--black-and-white] [--exclude-drawing-sheet] [--pages PAGE_LIST] INPUT_FILE
```

位置参数：

INPUT_FILE	要导出的原理图文件。
------------	------------

可选参数：

<code>-h, --help</code>	显示 DXF 文件导出命令的帮助信息。
<code>-o &lt;output dir&gt;, --output &lt;output dir&gt;</code>	导出文件的目标文件夹。如果不使用此参数，文件将被导出到当前目录。
<code>--drawing-sheet &lt;sheet path&gt;</code>	在绘制中使用的图纸路径，这将覆盖原理图文件中指定的图纸。
<code>-D &lt;variable name&gt;=&lt;value&gt;, --define-var &lt;variable_name&gt;=&lt;value&gt;</code>	添加或覆盖工程变量定义。可以多次使用来定义多个变量。
<code>-t &lt;theme name&gt;, --theme &lt;theme name&gt;</code>	用于导出的主题名称。如果未给出主题，将使用原理图编辑器当前选择的主题。
<code>-b, --black-and-white</code>	将原理图以黑白模式导出。
<code>-e, --exclude-drawing-sheet</code>	不使用图纸直接绘制 DXF。
<code>--pages &lt;page list&gt;</code>	要导出的页面的逗号分隔列表。空白或未指定则意味着导出所有页面。为了绘制特定页面，将根图纸作为 INPUT_FILE 给出，并使用 <code>--pages</code> 参数指定所需的输出页面。

## 原理图 HPGL 导出

`sch export hpgl` 命令将原理图导出到笔式绘图仪的 HPGL 文件中。设计中的每个图纸都导出到其自己的文件中。

```
Usage: kicad-cli sch export hpgl [--help] [--output OUTPUT_DIR] [--drawing-sheet SHEET_PATH] [--define-var KEY=VALUE] [--exclude-drawing-sheet] [--pages PAGE_LIST] [--pen-size PEN_SIZE] [--origin ORIGIN] INPUT_FILE
```

位置参数：

INPUT_FILE	要导出的原理图文件。
------------	------------

可选参数：

<code>-h, --help</code>	显示 HPGL 文件导出命令的帮助信息。
<code>-o &lt;output dir&gt;, --output &lt;output dir&gt;</code>	导出文件的目标文件夹。如果不使用此参数，文件将被导出到当前工作目录。
<code>--drawing-sheet &lt;sheet path&gt;</code>	用于绘制图纸路径，这将覆盖原理图文件中指定的图纸。
<code>-D &lt;variable name&gt;=&lt;value&gt;, --define-var &lt;variable_name&gt;=&lt;value&gt;</code>	添加或覆盖工程中的变量定义。可多次使用以定义多个变量。
<code>-e, --exclude-drawing-sheet</code>	不使用图纸背景，仅绘制 HPGL 数据。
<code>--pages &lt;page list&gt;</code>	要导出的页面列表，由逗号分隔。留空或不指定则默认导出所有页面。若要绘制特定页面，请将根图纸作为 INPUT_FILE 并使用 <code>--pages</code> 参数指定所需输出的页面。
<code>-p &lt;pen size&gt;, --pen-size &lt;pen size&gt;</code>	设置笔触宽度。默认笔宽为 0.5 mm。
<code>-r &lt;origin&gt;, --origin &lt;origin&gt;</code>	设置绘图仪原点和比例。可选值为 0、1（默认）、2 或 3。0 将原点设置在左下角并使用绘图仪单位；1 将原点设置在中心并使用绘图仪单位；2 按页面比例缩放；3 按页面内内容进行缩放。

## 原理图网表导出

`sch export netlist` 命令以 [各种格式](#) 格式导出网表。

用法：`kicad-cli sch export netlist [--help] [--output OUTPUT_FILE] [--format FORMAT] INPUT_FILE`

位置参数：

INPUT_FILE	要导出的原理图文件。
------------	------------

可选参数：

<code>-h, --help</code>	显示网络表导出命令的帮助信息。
<code>-o &lt;output filename&gt;, --output &lt;output filename&gt;</code>	指定输出文件的名称。如果不使用此参数，输出文件名将与输入文件相同，但文件扩展名会更改为 <code>.net</code> 。
<code>-f &lt;format&gt;, --format &lt;format&gt;</code>	网络表输出的格式。可选项包括 <code>kicadsexpr</code> （默认）、 <code>kicadxml</code> 、 <code>cadstar</code> 、 <code>orcadpcb2</code> 、 <code>spice</code> 或 <code>spicemodel</code> 。

## 原理图 PDF 导出

`sch export pdf` 命令将原理图导出为 PDF 文件。设计中的每张图纸都会导出到 PDF 文件中的单独页面。

Usage: `kicad-cli sch export pdf [--help] [--output OUTPUT_FILE] [--drawing-sheet SHEET_PATH] [--define-var KEY=VALUE] [--theme THEME_NAME] [--black-and-white] [--exclude-drawing-sheet] [--`

```
exclude-pdf-property-popups] [--no-background-color] [--pages PAGE_LIST] INPUT_FILE
```

位置参数：

INPUT_FILE	要导出的原理图文件。
------------	------------

可选参数：

<code>-h, --help</code>	显示 PDF 文件导出命令的帮助信息。
<code>-o &lt;output filename&gt;, --output &lt;output filename&gt;</code>	指定输出文件的名称。如果不使用此参数，输出文件名将与输入文件相同，但文件扩展名会更改为 .pdf。
<code>--drawing-sheet &lt;sheet path&gt;</code>	指定在绘制中使用的图纸路径，覆盖原理图文件中指定的图纸。
<code>-D &lt;variable name&gt;=&lt;value&gt;, --define-var &lt;variable_name&gt;=&lt;value&gt;</code>	添加或覆盖工程变量定义。可以多次使用来定义多个变量。
<code>-t &lt;theme name&gt;, --theme &lt;theme name&gt;</code>	指定用于导出的主题名称。如果未给定主题，则使用原理图编辑器当前选择的主题。
<code>-b, --black-and-white</code>	以黑白模式导出原理图。
<code>-e, --exclude-drawing-sheet</code>	导出 PDF 时不包含图纸。
<code>--exclude-pdf-property-popups</code>	不在 PDF 中生成属性弹窗。
<code>-n, --no-background-color</code>	不使用背景色导出原理图，无论所选主题如何。
<code>--pages &lt;page list&gt;</code>	要导出的页面列表，以逗号分隔。空白或未指定则意味着所有页面。为了导出特定页面，将根图纸作为 INPUT_FILE，并使用 --pages 参数指定所需的输出页面。

## 原理图 PostScript 导出

`sch export ps` 命令将原理图导出到 PostScript 文件。设计中的每个图纸都导出到其自己的文件中。

```
Usage: kicad-cli sch export ps [--help] [--output OUTPUT_DIR] [--drawing-sheet SHEET_PATH] [--define-var KEY=VALUE] [--theme THEME_NAME] [--black-and-white] [--exclude-drawing-sheet] [--no-background-color] [--pages PAGE_LIST] INPUT_FILE
```

位置参数：

INPUT_DIR	要导出的原理图文件。
-----------	------------

可选参数：



<code>-h, --help</code>	显示 PS 文件导出命令的帮助信息。
<code>-o &lt;output dir&gt;, --output &lt;output dir&gt;</code>	指定导出文件的目标文件夹。如果不使用此参数，文件将被导出到当前目录。
<code>--drawing-sheet &lt;sheet path&gt;</code>	指定绘制中要使用的图纸路径，覆盖原理图文件中指定的图纸。
<code>-D &lt;variable name&gt;=&lt;value&gt;, --define-var &lt;variable_name&gt;=&lt;value&gt;</code>	添加或覆盖工程变量定义。可多次使用来定义多个变量。
<code>-t &lt;theme name&gt;, --theme &lt;theme name&gt;</code>	指定用于导出的主题名称。若不给出主题，则使用原理图编辑器当前选定的主题。
<code>-b, --black-and-white</code>	以黑白模式导出原理图。
<code>-e, --exclude-drawing-sheet</code>	绘制 PS 文件时排除图纸。
<code>-n, --no-background-color</code>	导出原理图时不使用背景色，不论主题设置。
<code>--pages &lt;page list&gt;</code>	要导出的页面列表，以逗号分隔。留空或未指定则表示所有页面。为了导出特定页面，将根图纸作为 <code>INPUT_FILE</code> ，并利用 <code>--pages</code> 参数指定所需导出的页面。

## 原理图物料清单导出（旧版 BOM 脚本）

`sch export python-bom` 命令从原理图中导出 XML BOM 文件。然后，可以使用自定义脚本或外部参照中描述的本脚本之一将 XML BOM 文件处理为所需的 BOM 格式：[原理图 BOM 导出文档](#)。

用法：`kicad-cli sch export python-bom [--help] [--output OUTPUT_FILE] INPUT_FILE`

位置参数：

<code>INPUT_FILE</code>	要导出的原理图文件。
-------------------------	------------

可选参数：

<code>-h, --help</code>	显示 BOM 导出命令的帮助信息。
<code>-o &lt;output filename&gt;, --output &lt;output filename&gt;</code>	指定输出文件的文件名。如果不使用此参数，输出文件名将默认与输入文件相同，仅在原文件名后添加 <code>-bom.xml</code> 后缀及文件扩展名。

## 原理图 SVG 导出

`sch export svg` 命令将原理图导出为 SVG 文件。设计中的每个图纸都导出到其自己的文件中。

Usage: `kicad-cli sch export svg [--help] [--output OUTPUT_DIR] [--drawing-sheet SHEET_PATH] [--define-var KEY=VALUE] [--theme THEME_NAME] [--black-and-white] [--exclude-drawing-sheet] [--no-background-color] [--pages PAGE_LIST] INPUT_FILE`

位置参数：

INPUT_FILE	要导出的原理图文件。
------------	------------

可选参数：

-h, --help	显示 SVG 文件导出命令的帮助信息。
-o <output dir>, --output <output dir>	指定导出文件的目标文件夹。如果不使用此参数，文件将被导出到当前工作目录。
--drawing-sheet <sheet path>	指定用于绘制的图纸路径，覆盖原理图文件中指定的图纸设置。
-D <variable name>=<value>, --define-var <variable_name>=<value>	添加或覆盖工程变量定义。可多次使用以定义多个变量。
-t <theme name>, --theme <theme name>	指定用于导出的主题名称。若未指定主题，则使用原理图编辑器中当前选定的主题。
-b, --black-and-white	以黑白模式导出原理图。
-e, --exclude-drawing-sheet	在不使用图纸的情况下绘制 SVG。
-n, --no-background-color	导出原理图时不包含背景颜色，不受主题影响。
--pages <page list>	要导出的页面列表，由逗号分隔。留空或未指定则表示所有页面。要导出特定页面，请将根图纸作为 INPUT_FILE，并使用 --pages 参数指定所需的输出页面。

# 符号命令

`sym` 子命令将符号导出为另一种格式，或将符号库升级到 KiCad 符号文件格式的当前版本。

## 符号导出

`sym export svg` 命令将指定库中的一个或多个符号导出到 SVG 文件中。

Usage: `kicad-cli sym export svg [--help] [--output OUTPUT_DIR] [--theme THEME_NAME] [--symbol SYMBOL] [--black-and-white] [--include-hidden-pins] [--include-hidden-fields] INPUT_FILE`

位置参数：

INPUT_FILE	用于导出的符号库文件。
------------	-------------

可选参数：

<code>-h, --help</code>	显示符号 SVG 导出命令的帮助信息。
<code>-o &lt;output dir&gt;, --output &lt;output dir&gt;</code>	设置导出文件的输出文件夹。如果未使用此参数，文件将被导出到当前目录。
<code>-t &lt;theme name&gt;, --theme &lt;theme name&gt;</code>	指定用于导出的主题名称。如果没有给出主题，则使用符号编辑器中当前选择的主题。
<code>-s &lt;symbol name&gt;, --symbol &lt;symbol name&gt;</code>	指定从库中导出的特定符号。如果未使用此参数，则会导出库中的所有符号。
<code>--black-and-white</code>	以黑白模式导出符号。
<code>--include-hidden-pins</code>	在导出的 SVG 中包含隐藏引脚。
<code>--include-hidden-fields</code>	在导出的 SVG 中包含隐藏的符号字段。

## 符号升级

`sym upgrade` 命令将指定的符号库从传统符号格式升级到当前版本的 KiCad 的原生格式。如果输入库已采用当前文件格式，则不执行任何操作。

用法： `kicad-cli sym upgrade [--help] [--output OUTPUT_FILE] [--force] INPUT_FILE`

位置参数：

INPUT_FILE	要升级的符号库。
------------	----------

可选参数：

<code>-h, --help</code>	显示符号升级命令的帮助信息。
<code>-o &lt;output filename&gt;, --output &lt;output filename&gt;</code>	指定升级后符号库的输出文件名。如果不使用此参数，升级后的符号库将覆盖原来的库文件。
<code>--force</code>	即使输入的库已经是当前文件格式，也强制重新保存库文件。

# 版本命令

`version` 子命令打印 KiCad 版本。没有任何参数，它只打印版本号，例如 `7.0.7`。您可以使用 `--format` 参数以其他几种格式打印版本。

## NOTE

在 Gitlab 上提交错误报告或功能请求时，请使用 `kicad-cli version --format about` 获取要包含的版本信息。

用法： `kicad-cli version [--help] [--format VAR]`

可选参数：

<code>--format &lt;format&gt;</code>	版本号的格式。选项为 <code>plain</code> （默认）、 <code>commit</code> 或 <code>about</code> 。 <code>plain</code> 打印版本号（例如 <code>7.0.7</code> ），如果不使用 <code>--format</code> 参数，则为默认值。 <code>commit</code> 打印您正在使用的 KiCad 构建的 git commit 的哈希值。 <code>about</code> 打印完整版本信息，包括库版本和基本系统信息。您可以在 Bug 报告中 <a href="#">使用关于版本信息</a> 。
--------------------------------------	--