cādence™

Productivity Toolbox User Guide

Post Processing

February 2017

Table of Contents

Table of Figures

1 Overview

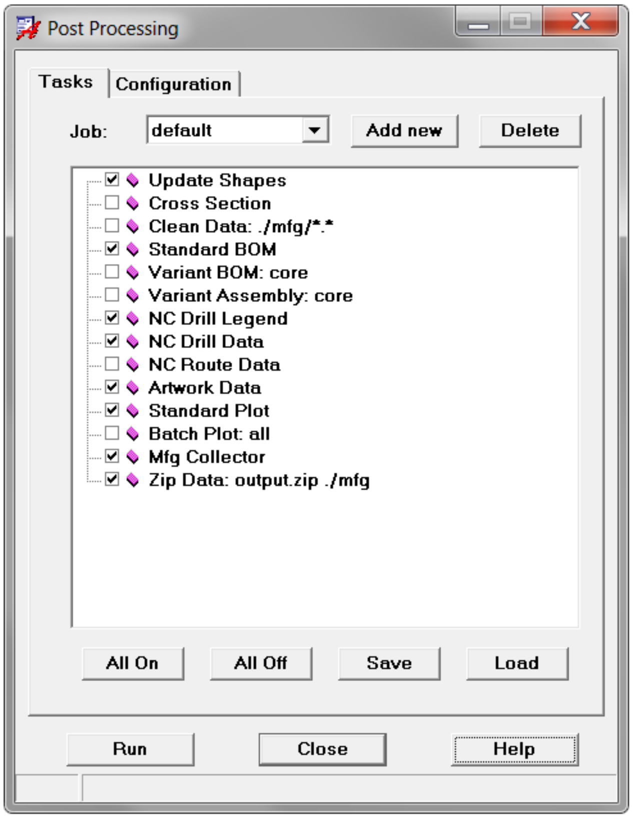
Once the schematic design and the PCB layout has been finished manufacturing data has to be generated. The amount and type of data to be generated differs significantly from one customer to another as not only drill and Gerber have to be generated but also various assembly drawings. The process of generating the manufacturing output usually takes a lot of time especially in the redesign process.

Post Processing is an application which acts as central cockpit from which manufacturing output tasks can be configured. Once configured the user only needs to “hit the button” so that manufacturing data will be generated in background. All information is stored in the database. If necessary the configuration can be saved to disk and loaded into another design.

Post Processing distinguishes between ***tasks*** and ***jobs***:

- A ***task*** represents a function from *PCB Editor* which generates data. A task can be a standard function from *PCB Editor* (e.g. Artwork generation, NC drill generation), a user defined macro or any other skill application (e.g. **Batch Plot**).

- A ***job*** represents a collection of tasks which are executed sequentially. For some reasons user might want to generate different data depending on the manufacturer chosen. Instead of enabling or disabling individual tasks, the user simply defines a new job with the corresponding tasks enabled. When generating data the user only needs to select the job and start the process.



Job: default

☑ Update Shapes
☐ Cross Section
☐ Clean Data: ./mfg/*.*
☑ Standard BOM
☐ Variant BOM: core
☐ Variant Assembly: core
☑ NC Drill Legend
☑ NC Drill Data
☐ NC Route Data
☑ Artwork Data
☑ Standard Plot
☐ Batch Plot: all
☑ Mfg Collector
☑ Zip Data: output.zip ./mfg

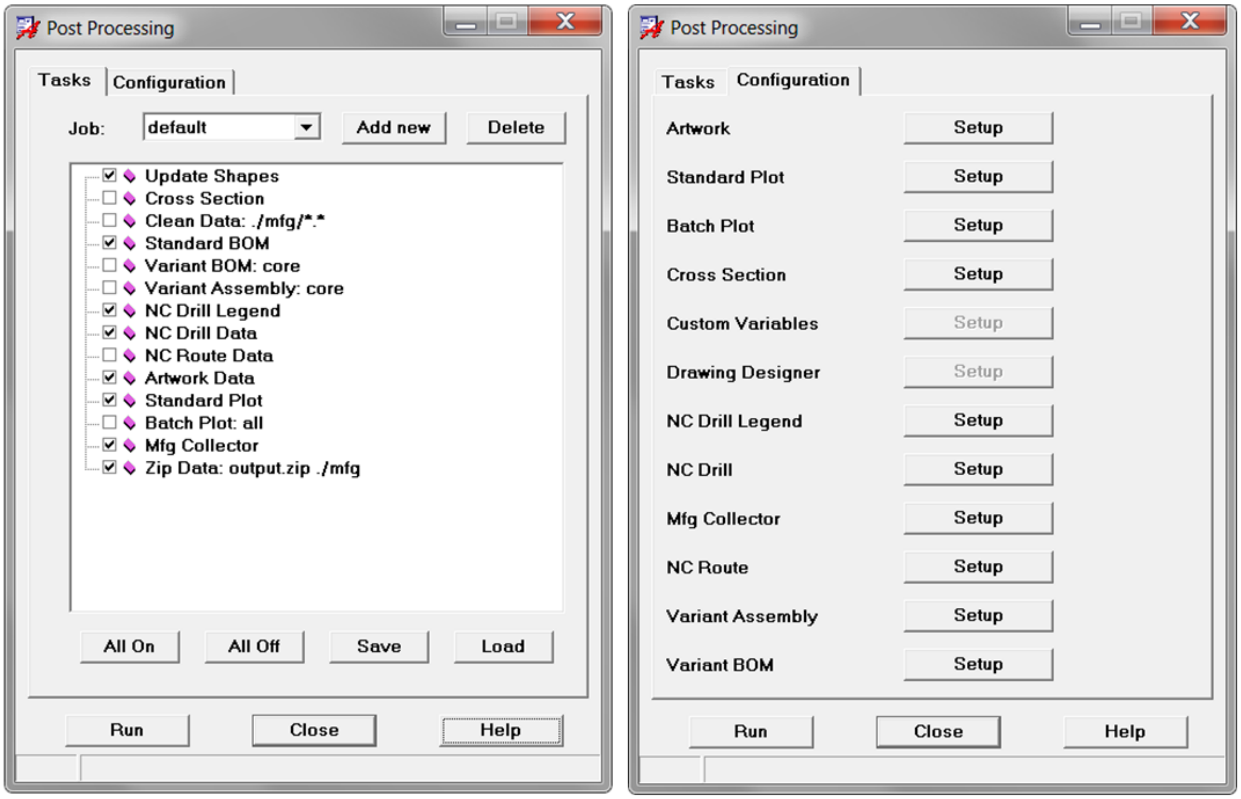
Figure 1: Post Processing

2 Use model

2.1 Launching the utility

Post Processing can be started from Pulldown menu or by entering `tbx postproc` in the console window.

Once the command has been launched a form appears, which is divided into two tabs.



Job: default

- ☑ Update Shapes
- □ Cross Section
- □ Clean Data: ./mfg/*.*
- ☑ Standard BOM
- □ Variant BOM: core
- □ Variant Assembly: core
- ☑ NC Drill Legend
- ☑ NC Drill Data
- □ NC Route Data
- ☑ Artwork Data
- ☑ Standard Plot
- □ Batch Plot: all
- ☑ Mfg Collector
- ☑ Zip Data: output.zip ./mfg

Figure 2: Post Processing form

2.2 Use model

The basic use model is as follows:

- Launch **Post Processing**
  When starting the tool on a new design, **Post Processing** will start with a default configuration. The active job is *default*. The tasks for this job are listed in the tree view.

- Enable or disable the tasks according to your needs. You may choose *All on* or *All off* to speed up selection.

- Several tasks (e.g. Artwork and NC Drill generation) require configuration if not already done. In order to do this, switch to the configuration tab and configure the corresponding task.

Note: When configuring standard functions from *PCB Editor* (e.g. *Artwork, NC Drill, NC Route* etc.) in the configuration tab, **Post Processing** will call the standard pull down menu.

- If necessary choose *Add new* in order to define a new job. Configure you tasks again for the new job.

- Choose the job from the drop down box for which you want to generate data.

- Click *Run*
  All tasks which are enabled will be executed in a sequential order.

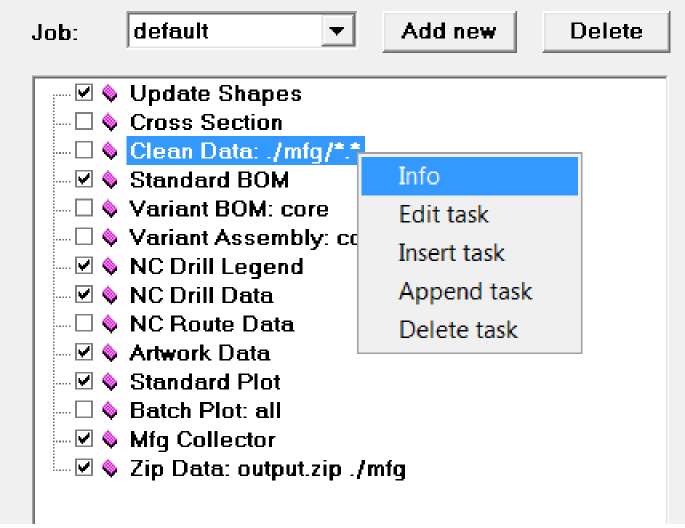
- Click *Close*
  Post Processing will close, however the configuration is stored in the database.

Note: You might use *Save* in order to save a configuration file to disk. Using the *Load* button you can also load such a configuration file into any *PCB Editor* database. This is useful for standardizing your outputs.

Note: You can use the console command `tbx postproc reset` if you want to delete the internal configuration from the database. For safety reasons a popup confirmer appears.

2.3 Task editing

The context menu inside the tree view provides additional functionality.



Job: default

☑ Update Shapes
☐ Cross Section
☐ Clean Data: ./mfg/*.*
☑ Standard BOM
☐ Variant BOM: core
☐ Variant Assembly: co
☑ NC Drill Legend
☑ NC Drill Data
☐ NC Route Data
☑ Artwork Data
☑ Standard Plot
☐ Batch Plot: all
☑ Mfg Collector
☑ Zip Data: output.zip ./mfg

Figure 3: Task editing using context menu

After selecting a task in the tree view and using the right mouse button, you can choose from:

- *Info selected*
  Gives some information about the task which is currently selected

- *Edit task*
  Allows editing of arguments for certain applications such as **Batch Plot**, **Variant BOM**, **Variant Assembly**, **Clean Data, Zip Data** etc.

- *Insert task*
  Will insert a new task before the selected task

- *Append task*
  Will append a new task after the selected task

- *Delete task*
  Will delete the task from the tree view which is currently selected

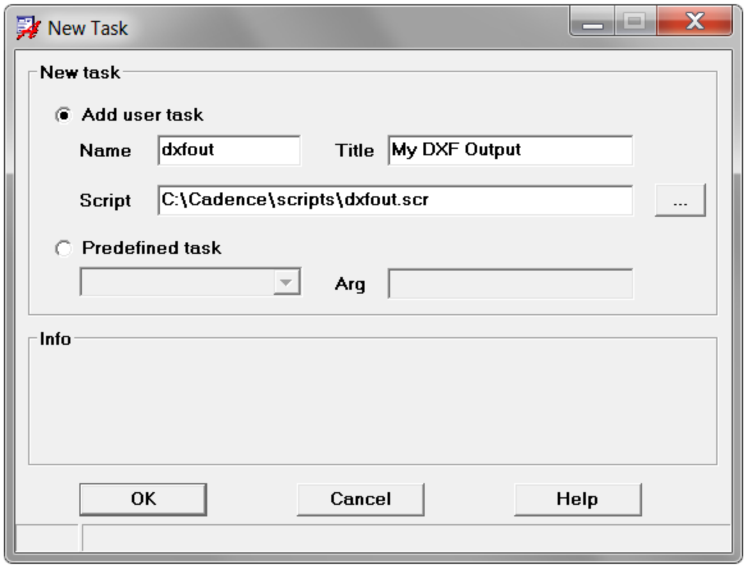
When choosing *Insert task* or *Append task,* another form appears in which task details can be specified. Two choices are possible:

- *Add user task*
  A user defined task is a task which is executed by a script. Usually those scripts have been recorded using the *File – Script* command. After selecting *Okay* the task will be added to the tree view. Three arguments have to be provided

  - ***Name***
    An abbreviation which serves as nick name for the task. Use characters and numbers only

  - ***Title***
    The value that appears in the tree view

  - ***Script***
    The path to the script which will be replayed during execution.



New Task

New task

● Add user task

Name: dxfout

Title: My DXF Output

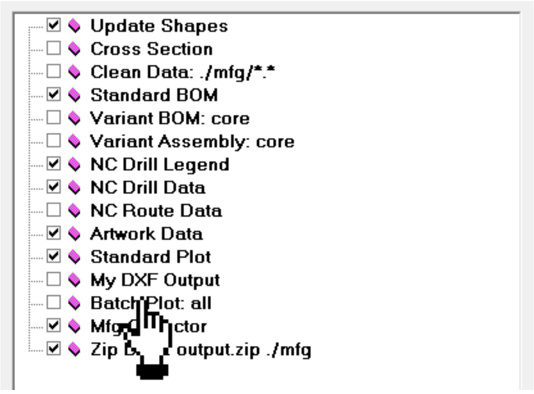
Script: C:\Cadence\scripts\dxfout.scr

○ Predefined task

Arg:

Info

Figure 4: Adding a user task

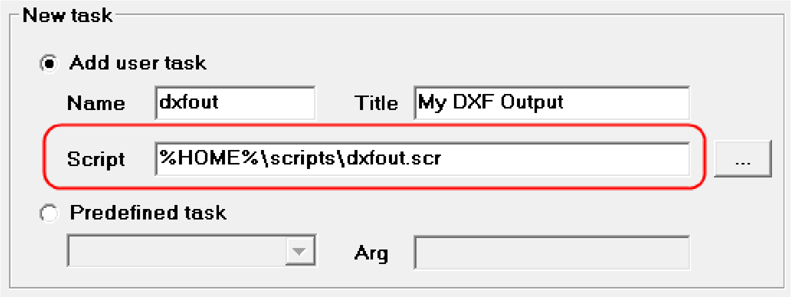


☑ Update Shapes
☐ Cross Section
☐ Clean Data: ./mfg/*.*
☑ Standard BOM
☐ Variant BOM: core
☐ Variant Assembly: core
☑ NC Drill Legend
☑ NC Drill Data
☐ NC Route Data
☑ Artwork Data
☑ Standard Plot
☐ My DXF Output
☐ Batch Plot: all
☑ Mfg ctor
☑ Zip output.zip ./mfg

Figure 5: User task display in tree view

Note: The content of the script will be simply embedded to the execution sequence without modification. Ensure that the script runs properly.

Note: You also use system variables in your path names in format `%var_name%` or `$var_name.` During execution variables will be expanded.



New task

● Add user task

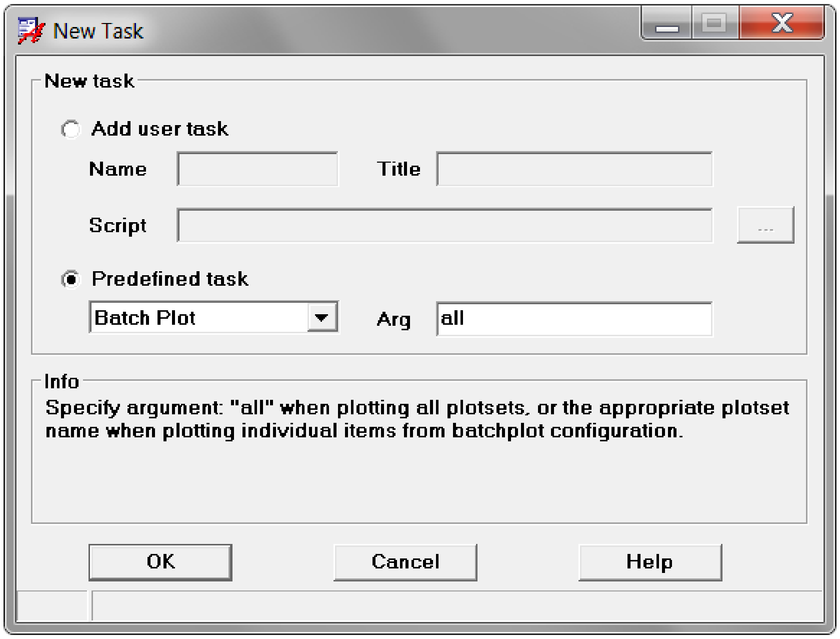
Name: dxfout   Title: My DXF Output

Script: %HOME%\scripts\dxfout.scr

○ Predefined task

Arg:

- *Predefined task*
  Predefined tasks are built-in functions in *PCB Editor* that is standard functions as well as skill procedures that were registered during startup. Select the appropriate task from the drop down menu. For some tasks an argument can be specified. In the *Info* section some hints are displayed for the selected task. After clicking *Okay* the task will be added to the tree view as shown above.



New Task

New task

○ Add user task

Name: Title:

Script:

● Predefined task

Batch Plot   Arg: all

Info

Specify argument: "all" when plotting all plotsets, or the appropriate plotset name when plotting individual items from batchplot configuration.

Figure 6: Adding a predefined task

3 Task description

The following section gives some detailed information about predefined tasks.

3.1 Update shape

This function is a built-in function of module **Post Processing** which updates all dynamic shapes in the design. Usually this is the first step before creating data. This task takes no arguments.

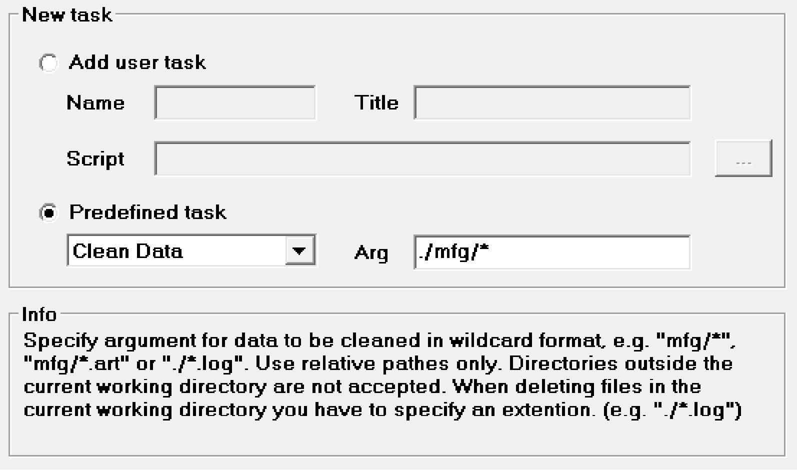
3.2 Cross Section Generator

Cross Section Generator is an application which gives customers the ability to generate a cross section drawing for documentation purposes. This function takes no arguments. For more details refer to **Cross Section Generator** user guide *share/pcb/toolbox/help/xsecgen.pdf* in the installation hierarchy.

Note: If the design does not contain a cross section symbol yet, **Post Processing** will place the cross section symbol in the upper left corner of the design.

3.3 Clean Data

In some cases directories need to be cleaned before new data is generated. **Clean data** is a built-in function in module **Post Processing** that performs delete operations based on arguments provided.



New task

○ Add user task

Name

Title

Script

● Predefined task

Clean Data

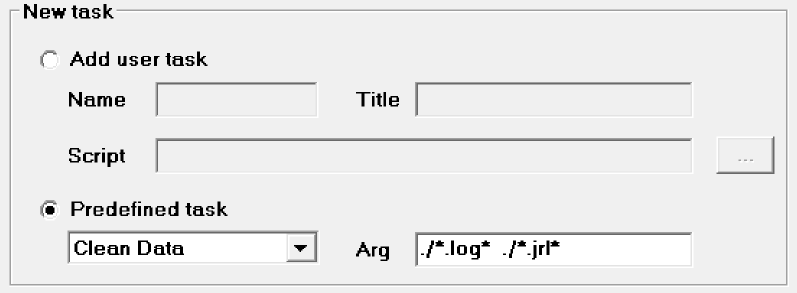
Arg ./mfg/*

Info

Specify argument for data to be cleaned in wildcard format, e.g. "mfg/*", "mfg/*.art" or "./*.log". Use relative pathes only. Directories outside the current working directory are not accepted. When deleting files in the current working directory you have to specify an extention. (e.g. "./*.log")

Figure 7: Clean Data task argument

Note: Multiple arguments can be specified if necessary. The following argument string will delete all log files and jrl files including backup versions (e.g. `allegro.jrl,1`) in the current working directory.



○ Add user task

Name:

Title:

Script:

● Predefined task

Clean Data

Arg: ./*.log* ./*.jrl*

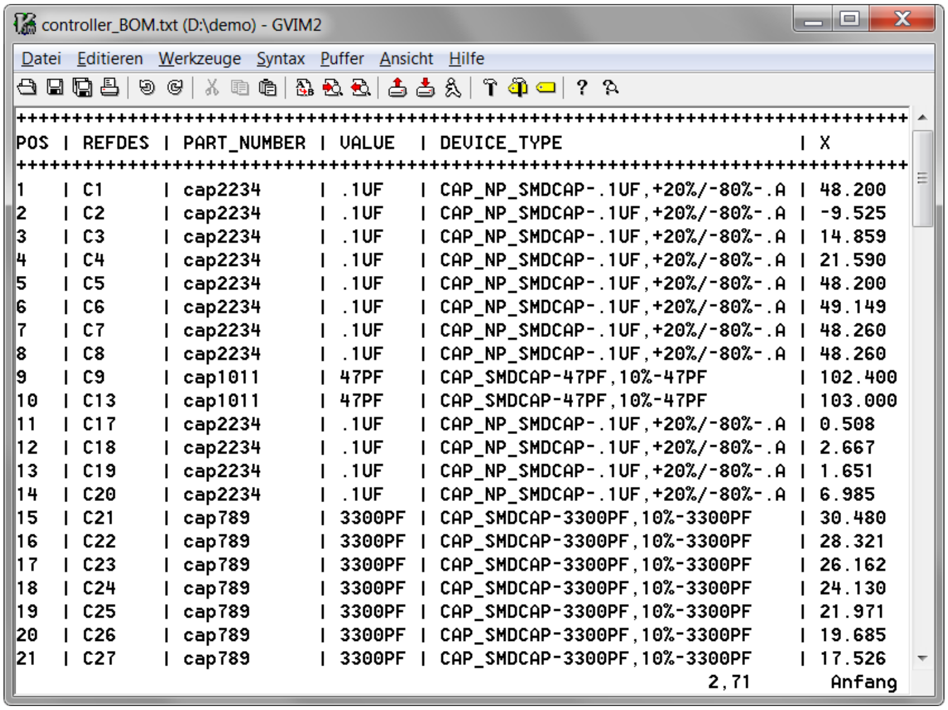
Figure 8: Clean Data multiple task arguments

Note: For safety reasons restrictions have been implemented

1. Directories to be cleaned have to be "inside" the local working directory pointers outside (e.g. `../../*.art` or any absolute path to another directory outside) are not allowed.
2. An argument `./*.*` or `*.*` or even `*` which specifies to delete all files in the current working directory is not allowed. When deleting files in the current working directory a file extension has to be defined at least (e.g. `./*.art`)

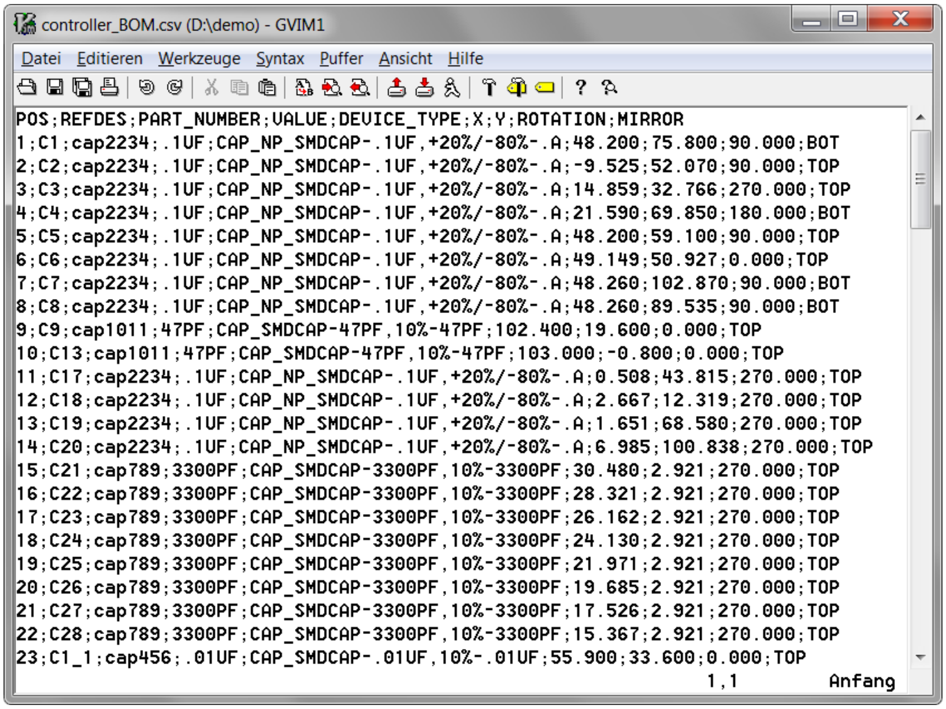
3.4 Standard BOM

Standard BOM is a built-in function of module **Post Processing** which generates a bill of material in csv and text format. The name of the files is `<design_name>_BOM.csv` and `<design_name>_BOM.txt` and will be written to the current working directory. This task takes no arguments. The content cannot be customized.



```
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
POS  | REFDES  | PART_NUMBER  | VALUE   | DEVICE_TYPE                      | X
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
1    | C1      | cap2234      | .1UF    | CAP_NP_SMDCAP-.1UF,+20%/-80%-.A | 48.200
2    | C2      | cap2234      | .1UF    | CAP_NP_SMDCAP-.1UF,+20%/-80%-.A | -9.525
3    | C3      | cap2234      | .1UF    | CAP_NP_SMDCAP-.1UF,+20%/-80%-.A | 14.859
4    | C4      | cap2234      | .1UF    | CAP_NP_SMDCAP-.1UF,+20%/-80%-.A | 21.590
5    | C5      | cap2234      | .1UF    | CAP_NP_SMDCAP-.1UF,+20%/-80%-.A | 48.200
6    | C6      | cap2234      | .1UF    | CAP_NP_SMDCAP-.1UF,+20%/-80%-.A | 49.149
7    | C7      | cap2234      | .1UF    | CAP_NP_SMDCAP-.1UF,+20%/-80%-.A | 48.260
8    | C8      | cap2234      | .1UF    | CAP_NP_SMDCAP-.1UF,+20%/-80%-.A | 48.260
9    | C9      | cap1011      | 47PF    | CAP_SMDCAP-47PF,10%-47PF         | 102.400
10   | C13     | cap1011      | 47PF    | CAP_SMDCAP-47PF,10%-47PF         | 103.000
11   | C17     | cap2234      | .1UF    | CAP_NP_SMDCAP-.1UF,+20%/-80%-.A | 0.508
12   | C18     | cap2234      | .1UF    | CAP_NP_SMDCAP-.1UF,+20%/-80%-.A | 2.667
13   | C19     | cap2234      | .1UF    | CAP_NP_SMDCAP-.1UF,+20%/-80%-.A | 1.651
14   | C20     | cap2234      | .1UF    | CAP_NP_SMDCAP-.1UF,+20%/-80%-.A | 6.985
15   | C21     | cap789       | 3300PF  | CAP_SMDCAP-3300PF,10%-3300PF     | 30.480
16   | C22     | cap789       | 3300PF  | CAP_SMDCAP-3300PF,10%-3300PF     | 28.321
17   | C23     | cap789       | 3300PF  | CAP_SMDCAP-3300PF,10%-3300PF     | 26.162
18   | C24     | cap789       | 3300PF  | CAP_SMDCAP-3300PF,10%-3300PF     | 24.130
19   | C25     | cap789       | 3300PF  | CAP_SMDCAP-3300PF,10%-3300PF     | 21.971
20   | C26     | cap789       | 3300PF  | CAP_SMDCAP-3300PF,10%-3300PF     | 19.685
21   | C27     | cap789       | 3300PF  | CAP_SMDCAP-3300PF,10%-3300PF     | 17.526
                                                               2,71        Anfang
```

Figure 9: Standard BOM report formatted text


controller_BOM.csv (D:\demo) - GVIM1
Datei Editieren Werkzeuge Syntax Puffer Ansicht Hilfe

```
POS;REFDES;PART_NUMBER;VALUE;DEVICE_TYPE;X;Y;ROTATION;MIRROR
1;C1;cap2234;.1UF;CAP_NP_SMDCAP-.1UF,+20%/-80%-.A;48.200;75.800;90.000;BOT
2;C2;cap2234;.1UF;CAP_NP_SMDCAP-.1UF,+20%/-80%-.A;-9.525;52.070;90.000;TOP
3;C3;cap2234;.1UF;CAP_NP_SMDCAP-.1UF,+20%/-80%-.A;14.859;32.766;270.000;TOP
4;C4;cap2234;.1UF;CAP_NP_SMDCAP-.1UF,+20%/-80%-.A;21.590;69.850;180.000;BOT
5;C5;cap2234;.1UF;CAP_NP_SMDCAP-.1UF,+20%/-80%-.A;48.200;59.100;90.000;TOP
6;C6;cap2234;.1UF;CAP_NP_SMDCAP-.1UF,+20%/-80%-.A;49.149;50.927;0.000;TOP
7;C7;cap2234;.1UF;CAP_NP_SMDCAP-.1UF,+20%/-80%-.A;48.260;102.870;90.000;BOT
8;C8;cap2234;.1UF;CAP_NP_SMDCAP-.1UF,+20%/-80%-.A;48.260;89.535;90.000;BOT
9;C9;cap1011;47PF;CAP_SMDCAP-47PF,10%-47PF;102.400;19.600;0.000;TOP
10;C13;cap1011;47PF;CAP_SMDCAP-47PF,10%-47PF;103.000;-0.800;0.000;TOP
11;C17;cap2234;.1UF;CAP_NP_SMDCAP-.1UF,+20%/-80%-.A;0.508;43.815;270.000;TOP
12;C18;cap2234;.1UF;CAP_NP_SMDCAP-.1UF,+20%/-80%-.A;2.667;12.319;270.000;TOP
13;C19;cap2234;.1UF;CAP_NP_SMDCAP-.1UF,+20%/-80%-.A;1.651;68.580;270.000;TOP
14;C20;cap2234;.1UF;CAP_NP_SMDCAP-.1UF,+20%/-80%-.A;6.985;100.838;270.000;TOP
15;C21;cap789;3300PF;CAP_SMDCAP-3300PF,10%-3300PF;30.480;2.921;270.000;TOP
16;C22;cap789;3300PF;CAP_SMDCAP-3300PF,10%-3300PF;28.321;2.921;270.000;TOP
17;C23;cap789;3300PF;CAP_SMDCAP-3300PF,10%-3300PF;26.162;2.921;270.000;TOP
18;C24;cap789;3300PF;CAP_SMDCAP-3300PF,10%-3300PF;24.130;2.921;270.000;TOP
19;C25;cap789;3300PF;CAP_SMDCAP-3300PF,10%-3300PF;21.971;2.921;270.000;TOP
20;C26;cap789;3300PF;CAP_SMDCAP-3300PF,10%-3300PF;19.685;2.921;270.000;TOP
21;C27;cap789;3300PF;CAP_SMDCAP-3300PF,10%-3300PF;17.526;2.921;270.000;TOP
22;C28;cap789;3300PF;CAP_SMDCAP-3300PF,10%-3300PF;15.367;2.921;270.000;TOP
23;C1_1;cap456;.01UF;CAP_SMDCAP-.01UF,10%-.01UF;55.900;33.600;0.000;TOP
                                                              1,1          Anfang
```

Figure 10: Standard BOM report CSV format

3.5 Variant BOM

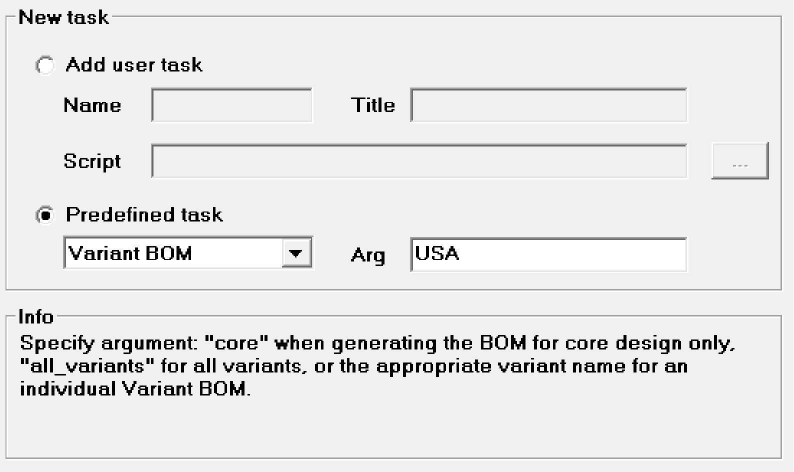
Both schematic tools – *Design Entry CIS* and *Design Entry HDL* – offer the ability to create variants. On the layout side the variant information can be taken to create variant specific assembly drawings. Variant Reports (BOM) can be generated from Schematic and from Layout, but until now it’s not possible to include variant information into Pick&Place reports. This leads to a lack of flexibility, when variant Pick&Place data need to be generated. **Variant BOM** is an application which allows the user to create advanced Pick&Place reports with the ability to take variant information into account. For more details refer to **Variant BOM** user guide which can be found in the installation hierarchy under *share/pcb/toolbox/help/varibom.pdf*

Arguments have to be provided:

- **`core`**
  Generates advanced reports for the master (core) design

- **`all_variants`**
  Generates advanced reports for all variants in the design

- **`<variant_name>`**
  Generates advanced reports for a particular variant name defined in `variants.lst` file



○ Add user task

● Predefined task

Variant BOM ▼ Arg USA

Specify argument: "core" when generating the BOM for core design only, "all_variants" for all variants, or the appropriate variant name for an individual Variant BOM.

Figure 11: Variant BOM task argument

3.6 Variant Assembly

Both schematic tools – *Design Entry CIS* and *Design Entry HDL* – offer the ability to create variants. The variant information is passed to PCB Editor using a file called `variants.lst`. In *PCB Editor* the variant information can be taken to create variant assembly views.

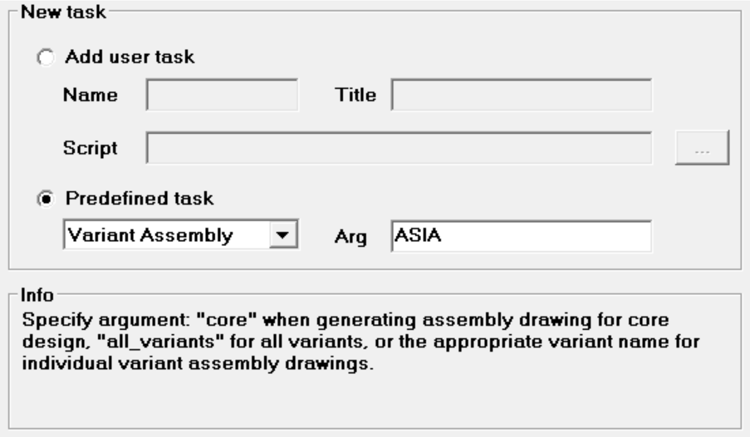
Variant Assembly is an application which gives customers some more flexibility when creating variant assembly views (e.g. automatic mirror of BOTTOM view side by side to TOP view). For more details refer to **Variant Assembly** user guide *share/pcb/toolbox/help/variassy.pdf* in the installation hierarchy.

Arguments have to be provided:

- **`core`**
  Generates an assembly view for the master (core) design

- **`all_variants`**
  Generates assembly views for all variants in the design

- **`<variant_name>`**
  Generates an assembly view for a particular variant name defined in `variants.lst` file



○ Add user task

● Predefined task

Variant Assembly ▼ Arg ASIA

Info

Specify argument: "core" when generating assembly drawing for core design, "all_variants" for all variants, or the appropriate variant name for individual variant assembly drawings.

Figure 12: Variant Assembly task argument

3.7 NC drill legend

This task is a standard function from PCB Editor (*Manufacture – NC – Drill Legend*) which lets you create different types of drill legend tables, which sort hole sizes and map drill figures or text symbols to each drill bit size. When enabled the Drill legend command is executed in batch. This task takes no arguments.

3.8 NC Drill data

This task is a standard function from *PCB Editor* (*Manufacture – NC – NC Drill*) which generates customized NC drill output files. When this task is enabled NC Drill data will be generated in batch. This task takes no arguments.

3.9 NC Route data

This task is a standard function from *PCB Editor* (*Manufacture – NC – NC Route*) which generates customized NC route output files. When this task is enabled NC Route data will be generated in batch. This task takes no arguments.

3.10 Artwork data

This task is a standard function from *PCB Editor* (*Manufacture – Artwork*) which generates photoplot film files. When this task is enabled film data will be generated in batch. This task takes no arguments.

3.11 Standard Plot

Standard Plot is a built-in function of module **Post Processing** which generates a combined PDF file of all artwork film control records in the design. This task takes no arguments. The content cannot be customized. However, in order to use this function you have to setup a PDF writer first. For configuration details refer to chapter 2 of *share/pcb/toolbox/help/batchplot.pdf* in the installation hierarchy.

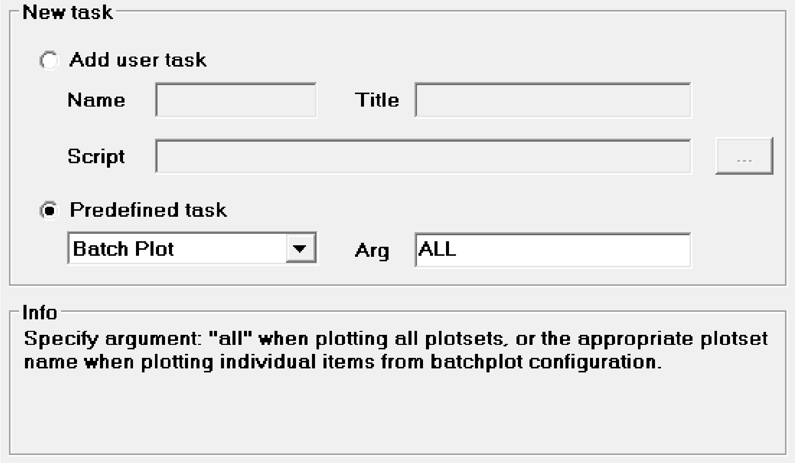
3.12 Batch Plot

In many cases customers need to create plot files (e.g. in PDF format) in batch and/or combine those into a single PDF file. The plotting procedure in PCB Editor is based on the Windows plotting mechanism, which is rather interactive. For each plot the user needs to adjust the visibility settings in PCB Editor (or use views), and then plot each view directly to the plotter or into a file. There are various PDF writers available (such as *Adobe PDF* or *FreePDF XP*), which are virtual plotters, but the process of creating each plot file individually does not change. The **Batch Plot** utility allows the user to create a couple of PDF documents in one step. In addition it allows the user to combine several files into one document. For more details refer to **Batch Plot** user guide which can be found under *share/pcb/toolbox/help/batchplot.pdf* in the installation hierarchy.

Arguments have to be provided:

- **`all`**
  PDF files will be generated for each plotset defined in the configuration

- **`<plotset_name>`**
  A PDF file will be generated only for the plotset specified by argument <plotset_name>



● Predefined task

Batch Plot

Arg ALL

Specify argument: "all" when plotting all plotsets, or the appropriate plotset name when plotting individual items from batchplot configuration.

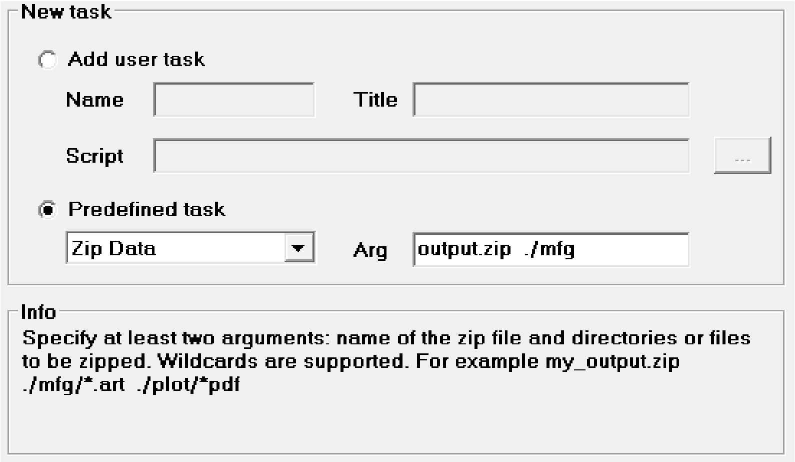
Figure 13: Batch Plot task argument

3.13 Mfg Collector

Once all files have been generated customers often need to rename and move files for various purposes. For example each artwork file needs to be suffixed by a PART_NUMBER due to PLM requirements. Furthermore the file extension might need to be changed form **`.art`** to **`.gbr`**
MFG Collector is an application which lets you “collect” manufacturing data by moving files from one location to another including powerful renaming operations. The operations to be executed are specified by rules. For more details refer to **Mfg Collector** user guide *share/pcb/toolbox/help/mfgcollect.pdf* in the installation hierarchy.

3.14 Zip Data

Finally data might need to zipped into a single file. **Zip Data** is a built-in function in module **Post Processing** that performs a zip operation based on arguments provided.



New task

○ Add user task

Name

Title

Script

● Predefined task

Zip Data

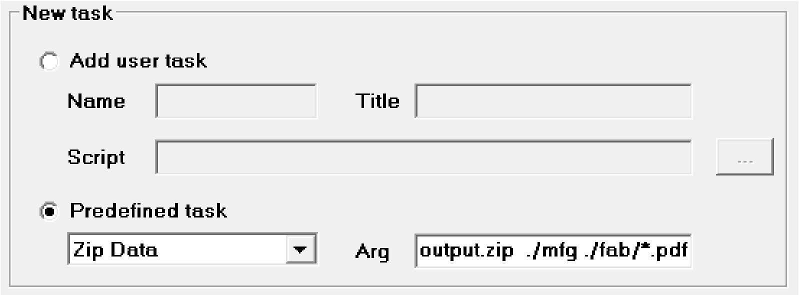
Arg: output.zip ./mfg

Info

Specify at least two arguments: name of the zip file and directories or files to be zipped. Wildcards are supported. For example my_output.zip ./mfg/*.art ./plot/*pdf

Figure 14: Zip Data task arguments

Zip Data takes at least two arguments. The first argument always refers to the name of the zip file, while the remaining arguments specify the files or directories to be included. Wildcards can be used if necessary. The following example will zip directory `./mfg` and all pdf files from subdirectory `fab` into a single file `output.zip`.



New task

○ Add user task

Name

Title

Script

● Predefined task

Zip Data

Arg: output.zip ./mfg ./fab/*.pdf

Figure 15: Zip Data multiple task arguments