# CadPack

# Import from ODB++

Software tool for import from ODB++ Cad format

## Technical Info

**Version :** 2
**Code :** 81190637.162

**SPEA SpA**
16, Via Torino
10088 Volpiano - Italy
Tel.: + 39 011 9825 400
Fax: + 39 011 9825 405
E-mail: info@spea.com
Web: www.spea.com

# Contents

# Introduction

CAD files are the base for the automatic generation of test program for InCircuit of any technology.

In order to generate the ICT test program in a short time and without errors, both Bed of Nails and Flying Probe testers require the circuit information available in CAD format.

The “Import from ODB++” CAD import driver allows to import the data present in the ODB++ CAD file and convert them in the SPEA Board data format.

***Conventions, symbols and abbreviations***

In the document, the  symbol is used to highlight information or notes useful to the reader.



This manual can be updated in accordance with the evolution of the system and associated software.
It may contain preliminary contents or it may not be entirely updated with the latest versions used in the system.

Any remarks on errors and imperfections, or suggestions, can be addressed to:

**SPEA SpA**
Ufficio Documentazione
16, Via Torino
10088 Volpiano – Italy
Tel.: +39 011 9825400
Fax: +39 011 9825405
Email: info@spea.com
Web: www.spea.com

# 1. ODB++ file data

With the “ODB++ CAD files” words we refer to the output information generated by the ODB++ programs for the electrical diagrams design and PCB development, used to develop a test application (test program and adapter design).

Information stored in the “ODB++ CAD files” concern an electronic board and can be used by an appropriate program to generate a test program and its test adapter design (Bed of Nails or list of movement for Flying Probes).

Information can be grouped in 4 different categories and typically concern the printed circuit:

- **Part List**
  It is the list of all used devices, it must contain: devices drawing reference, part numbers, value, tolerances, device type, etc.
- **Net list**
  It is also called wiring list, containing device interconnection data; basically it is presentation of the electrical diagram.
- **Coordinate and access list**
  It is the list containing the devices coordinates, concerning their barycentre and pins.
- **Wiring and Routing list**
  It is the list containing the path of the Net tracks in the PCB.

For the import of the information above mentioned SPEA has developed the specific program for the translation, stored in a specified format, to its common data bank called “Board Data”.
The name of this type of program is “CAD import driver”.

For the required information, see the list in the following paragraphs.

## 1.1 Part List

The Part List is an ASCII text file, containing the list of all the parts used to assemble the board; sometimes it can be called **Bill of Material** (BOM).

In the Part List, all information concerning the mounted and not mounted parts must be present.
For every part the following information must be defined:

| Information | Description |
|---|---|
| **Drawing Reference** | Reference designator (e.g. U10, R105, D23, etc.). |
| **Part Number** | Device code (e.g. 132549.012, C4QW08, 001-58-AA, etc.). |
| **Value** | Device value (e.g. 10KΩ, 10µF, 1mH, etc.). |
| **Tolerance** | Positive and negative device tolerances (e.g. 1%, 5%, etc.). |
| **Mounting side** | The legal values for this item can be:<br>♦ **Top** (Component side)<br>♦ **Bottom** (Soldering side)<br>♦ **Not mounted Top**<br>♦ **Not mounted Bottom** |
| **Rotation** [1] | Device mounting rotation angle (e.g. 0°, 180°, etc.). |
| **Dimensions** [1] | Device dimensions. |
| **Case code** [1] | Device package (case) code. |

*Table 1 – Part List*

[1] **Optional data (not yet managed)**

## 1.2 Net List

The Net List is an ASCII text file containing the device interconnection data; it is also called wiring list. This list must contain the interconnection between devices, including pad and via.

Basically, it is the representation of the electrical diagrams.

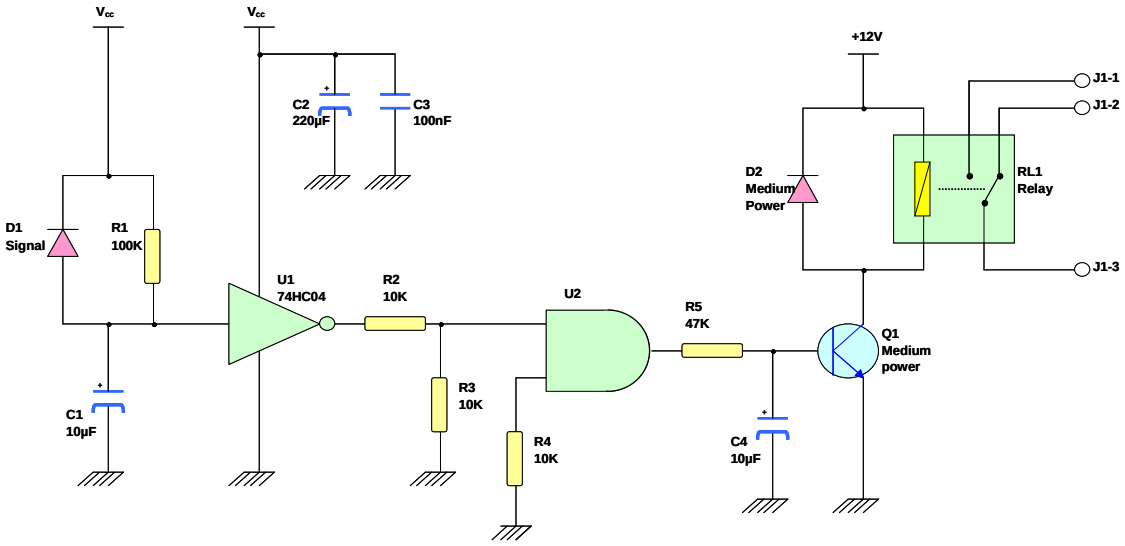


*Figure 1 – Electrical diagram example*

For every net the following information must be defined:

| Information | Description |
|---|---|
| **Net name** | Net identifier (e.g. +5V, RESET, A01, etc.). |
| **Drawing reference** | Reference designator of the device connected to the net (e.g. U10, R105, D23, etc.). |
| **Pin name** | Name of the device pin connected to the net (e.g. 1, 15, Anode, K, Negative, etc.). |
| **Pin access side** | Access side for the device pin, legal values are:<br>♦ **Top** (Device side access)<br>♦ **Bottom** (Soldering side access).<br>♦ **Not accessible**<br>♦ **All** (both top and bottom side access) |

*Table 2 – Net List*

## 1.3 Coordinates and access list

The Coordinates and access list is an ASCII text file containing the devices coordinates concerning their barycentre and pins. Below, the required information:

| Information | Description |
|---|---|
| **Drawing Reference** | Reference designator of the device connected to the net (e.g. U10, R105, D23, etc.). |
| **Pin name** | Name of the device pin connected to the net (e.g. 1, 15, Anode, K, Negative, etc.). |
| **Pin X position** | Pin X-coordinate. |
| **Pin Y position** | Pin Y-coordinate. |
| **X barycentre** [1] | Device X barycentre. |
| **Y barycentre** [1] | Device Y barycentre. |

*Table 3 – Coordinates and access list*

## 1.4 Wiring and Routing list

The Wiring and Routing list is an ASCII text file that contains all the coordinates of the Net tracks on the PCB and the link with the Net List. So the path of each net on the PCB is described in this file.

For every net the following information must be defined:

| Information | Description |
|---|---|
| **Net name** | Net identifier (e.g. +5V, RESET, A01, etc.). |
| **X Start** | Track segment start X-coordinate. |
| **Y Start** | Track segment start Y-coordinate. |
| **X End** | Track segment end X-coordinate. |
| **Y End** | Track segment end Y-coordinate. |
| **Width** | Net segment thickness. |
| **Layer** | Layer the segment belongs to. |

*Table 4 – Wiring and Routing list*

Example:

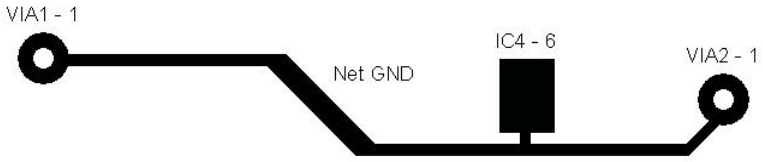


*Figure 2 – Net track example*

[1] **Optional data**

# 2. ODB++ file generalities

The ODB++ is the CAD/CAM format, capturing all CAD/EDA database, assembly and PCB fabrication knowledge in single, unified database.

The ODB++ database is a complex complete description of a printed circuit board that is often requested by board manufacturers in order to consolidate all of the information needed to produce and test the board in one job.

## 2.1 ODB++ job structure

A ODB++ job is represented by a self standing directory tree, which means the job tree can be transferred between computer systems without loss of data. All files in ODB++ are readable ASCII files.

A ODB++ job consists of a “Obdjob” directory composed of many subdirectories and files.


*Figure 3 – ODB++ job*

Of the various subdirectories, SPEA needs only consider the following folders that contribute to layout data:

- **Matrix**, it takes the form of an array in which the rows are the job layers and the columns are the job steps. The matrix contains for each row additional information such as the name, type, polarity and context. The matrix is important in defining the physical order of the layers and the relation of drill layers (through, blind, buried, etc.).
- **Steps**, it contains variuos sub-folders including the folder *layers*, which contais output for each layer enabled for plotting, as well as drill information and component information. The Steps are "containers" which essentially describe a multi-layer object: a PCB, a panel outline, a test coupon or similar item.
- **Layers**, it contains a series of layers folders. The layer folder contains the drawable “feature” file that are converted. The feature file lists the actual draws, flashes and surfaces that create the board.
- Special layers for components.

## 2.2 ODB++ job file name

The ODB++ job file name has to have the **.TGZ** or **.ZIP** extensions.


In order to import the ODB++ job in SPEA software correctly, it has to be organized with the following foders and files:

| Paths | Files |
|---|---|
| odbjob\matrix\ | *matrix* |
| odbjob\steps\<step name>\eda\ | *data* |
| odbjob\steps\<step name>\layers\comp_+_top\ | *component* |
| odbjob\steps\<step name>\layers\comp_+_bot\ | *component* |
| odbjob\steps\<step name>\layers\<layers name>\ | *features* |

*Table 5 – Odb++ job file organization*

# 3. ODB++ file format

## 3.1 Matrix file format

This is a partial extract of an example of a MATRIX file:

```
STEP {
    COL=1
    NAME=<STEP NAME>
}


LAYER {
    ROW=1
    CONTEXT=BOARD
    TYPE=COMPONENT
    NAME=COMP_+_TOP
    POLARITY=POSITIVE
    START_NAME=
    END_NAME=
    OLD_NAME=
}

LAYER {
    ROW=2
    CONTEXT=BOARD
    TYPE=SILK_SCREEN
    NAME=SST
    POLARITY=POSITIVE
    START_NAME=
    END_NAME=
    OLD_NAME=
}
...
```

## 3.2 Data file format

This is a partial extract of an example of a DATA file:

```
HDR Expedition PCB
LYR layer_12 layer_1 layer_9 layer_7 layer_3 layer_10 layer_4 layer_6 d_1_12 layer_2 layer_5
layer_8 layer_11 spt spb smt smb sst ssb ddt fab_drawing dimentions title v-scored_rails
fabrication_text
#
# NET 0
NET $NONE$
SNT TOP T 5 15
FID C 1 14803
FID C 13 54
FID C 15 54
SNT TOP T 5 42
FID C 1 14776
FID C 13 27
FID C 15 27
SNT TOP T 6 19
FID H 8 6670
FID C 1 14933
FID C 4 14427
FID C 6 16789
FID C 7 14774
FID C 3 15751
FID C 2 16390
FID C 5 13077
FID C 0 15842
FID C 15 184
FID C 16 150
...
```

## 3.3 Component file format

This is a partial extract of an example of a COMPONENT file:

```
#Component attribute names
#
@0 .comp_mount_type
@1 .comp_height

# CMP 0
CMP 60 -0.8999999 1.3 0.0 N R616 ERJ3GEYJ104V ;0=1
TOP 0 -0.9329998 1.3 0.0 N 1094 0 1
TOP 1 -0.8669999 1.3 0.0 N 27 0 2
#
# CMP 1
CMP 56 -5.7749999 5.15 180.0 N S6 CTS219-4LPST ;0=1,1=0.145000
TOP 0 -5.625 5.3194999 180.0 N 12 0 1
TOP 1 -5.725 5.3194999 180.0 N 12 1 2
#
# CMP 2
CMP 60 -5.9499999 7.275 270.0 N R137 ERJ3GEYJ102V ;0=1
TOP 0 -5.9499999 7.2420001 270.0 N 14 0 1
TOP 1 -5.9499999 7.308 270.0 N 414 0 2
#
# CMP 3
CMP 59 -0.8999999 1.075 0.0 N R611 ERJ2GEJ272X ;0=1
TOP 0 -0.9249998 1.075 0.0 N 1113 0 1
TOP 1 -0.8749999 1.075 0.0 N 26 0 2
#
...
```

## 3.4 Features file format

This is a partial extract of an example of a FEATURES file:

```
#Feature symbol names
#
$0 r5
$1 r6
$2 r9.84
$3 r9.843
$4 r10
$5 r12
$6 r15
$7 r20
$8 r25
$9 r35
$10 r40
$11 r50
$12 r60
$13 r75
$14 r86
$15 r100
$16 r200
$17 s39.37
$18 s75
$19 esd_strip_pad_1a
$20 esd_strip_pad_2_6u
$21 esd_strip_pad_3a
...
#
#Feature attribute names
#
@0 .nomenclature
@1 .geometry
@2 .string
@3 .pad_usage

#
#Feature attribute text strings
#
&0 v50
&1 v25t
&2 v20
&3 43x40
&4 44x96
&5 35x27
&6 18x63
&7 40h24
&8 16x70
&9 102x87
&10 Pad_Round_20
&11 56x51
...
#
#Layer features
#
L -3.1009999 4.4 -3.0244999 4.4 6 P 0
L -0.4649999 0.29051 -0.4649999 0.26451 6 P 0
L -0.4649999 0.26451 -0.4695099 0.26 6 P 0
L -0.4695099 0.26 -0.5502699 0.26 6 P 0
L -0.5502699 0.26 -0.5999999 0.30973 6 P 0
L -0.5999999 0.30973 -0.5999999 0.311024 6 P 0
L -4.1288799 5.85344 -4.1104999 5.83506 6 P 0
L -4.1104999 5.83506 -4.1104999 5.83379 6 P 0
L -4.1104999 5.83379 -4.0917099 5.815 6 P 0
L -4.0917099 5.815 -4.0312399 5.815 6 P 0
L -4.0312399 5.815 -4.0160099 5.83023 6 P 0
L -4.0160099 5.83023 -4.0160099 5.85177 6 P 0
L -4.0160099 5.85177 -4.0619999 5.89776 6 P 0
L -4.0619999 5.89776 -4.0619999 5.95 6 P 0
...
```