
Koala Documentation

Release 1.0.1

Chintalagiri Shashank

January 19, 2015

I	User Documentation	1
1	Installation	3
1.1	Setting up pyenv	3
1.2	Getting the Code	4
1.3	Setting up virtualenv	4
1.4	Installing the Dependencies	5
2	Conventions	7
2.1	gEDA Symbol Conventions	7
2.1.1	Symbol Attributes	7
2.1.2	Device Classes	8
2.1.3	Values	9
	General	9
	Resistors	10
	Capacitors	10
	Diodes	11
	Inductors	11
	Crystals	12
	Connectors	12
2.1.4	Component Groups	13
2.1.5	Symbol Status	13
2.1.6	Attribute Promotion	13
2.1.7	Script Dependencies	13
	conventions.electronics	13
	sourcing.electronics	14
	sourcing.digikey	14
II	Developer Documentation	15
3	gEDA Interface Module Documentation	17
3.1	gEDA ConfigsFile module documentation (gedaif.conf)	17
3.2	gEDA Project File module documentation (gedaif.proj)	17
3.3	gEDA BOM Parser module documentation (gedaif.bomparser)	17
3.4	gEDA gschem module documentation (gedaif.gschem)	17
3.5	gEDA gsymlib Module documentation (gedaif.gsymlib)	18
4	Entity Hub Modules Documentation	19
4.1	EntityHub Transforms Module documentation (entityhub.transforms)	19
4.2	EntityHub Maps Module documentation (entityhub.maps)	19

5	Boms Module documentation	21
5.1	Electronic Boms Module documentation (<code>boms.electronics</code>)	21
5.1.1	Module Summary:	21
5.1.2	Module Members:	21
6	Inventory Module documentation	25
6.1	Electronics Inventory module documentation (<code>inventory.electronics</code>)	25
6.2	Inventory Acquire Module documentation (<code>inventory.acquire</code>)	25
7	Sourcing Module documentation	27
7.1	Electronics Sourcing module documentation (<code>sourcing.electronics</code>)	27
7.2	Vendors module documentation (<code>sourcing.vendors</code>)	27
7.3	Digi-Key Sourcing Module documentation (<code>sourcing.digikey</code>)	28
7.4	CSIL Sourcing Module documentation (<code>sourcing.csil</code>)	29
8	Dox Module documentation	31
8.1	Production Dox module documentation (<code>dox.production</code>)	31
8.2	gEDA Project Dox module documentation (<code>dox.gedaproject</code>)	31
8.3	Dox Render module documentation (<code>dox.render</code>)	31
9	Utils Module documentation	33
9.1	Config Module Documentation (<code>utils.config</code>)	33
9.2	Currency module documentation (<code>utils.currency</code>)	33
9.3	Filesystem Utils Module Documentation (<code>utils.fsutils</code>)	33
9.4	LibreOffice Utils Module Documentation (<code>utils.ooutils</code>)	34
9.5	PDF Utils Module documentation (<code>utils.pdfutils</code>)	34
9.6	WWW Utils Module Documentation (<code>utils.wwwutils</code>)	35
	Python Module Index	37
	Index	39

Part I

User Documentation

INSTALLATION

For the moment, koala is not designed to be installed to the system (/usr, /opt, etc). Such installation scripts may be developed for later versions.

Setting up Koala is essentially a matter of cloning/checking out the appropriate git/svn repository and providing the software with a suitable environment which contains all the necessary dependencies. The instructions listed here are for Ubuntu, and should work as-is on most Debian based Linux distributions.

The use of `pyenv` and `virtualenv` for this is recommended but not necessary. The use of these tools should create a reasonable stable environment until any cross-version kinks are found and worked out.

Compatibility of the scripts has not been tested with any python version other than the `Python 2.7.6` which comes with `Ubuntu 14.04.1 LTS Trusty Tahr`. They will almost certainly not work with `Python 3.x` in their current form and may or may not work with older point releases of `Python 2.7`. While they may work, it's probably a good idea to avoid using Python versions < 2.7 with these scripts until a full set of unit tests can be prepared and run.

1.1 Setting up pyenv

`pyenv` is needed to easily set up multiple python versions on your computer.

See <http://davebehnke.com/python-pyenv-ubuntu.html> for a more detailed explanation.

1. Install Git:

```
sudo apt-get install git-core curl
```

3. Setup proxy, if any:

```
export http_proxy=http://user:pass@192.168.1.254:3128
export https_proxy=http://user:pass@192.168.1.254:3128
export ftp_proxy=http://user:pass@192.168.1.254:3128
```

4. Tell git to use `https://` instead of `git://` to get around proxy issues:

```
git config --global url."https://".insteadOf git://
```

5. Run the installer:

```
curl -L \
https://raw.githubusercontent.com/yyuu/pyenv-installer/master/bin/pyenv-installer \
| bash
```

6. Insert the following at the end of `~/ .bashrc`:

```
export PYENV_ROOT="${HOME}/.pyenv"
if [ -d "${PYENV_ROOT}" ]; then
    export PATH="${PYENV_ROOT}/bin:${PATH}"
    eval "$(pyenv init -)"
fi
```

7. Install Build Dependencies for Python 2.7:

```
sudo apt-get build-dep python2.7
sudo apt-get install build-essential wget \
    libreadline-dev libncurses5-dev libssl1.0.0 tk8.5-dev \
    zlib1g-dev liblzma-dev
```

8. Install Python 2.7.6:

Python 2.7.x, where $x \geq 6$, should be fine. $x < 6$ is untested. New features were introduced in 2.7.5, 2.7.6 that may be necessary for the scripts to run. If system python is 2.7.6 or better, pyenv isn't strictly necessary. However, to standardize the environment in the absence of cross-version testing and intelligent installation scripts, the use of a version-specified python version (as opposed to `system`) is recommended.

```
pyenv install 2.7.6
```

1.2 Getting the Code

The code can be obtained from the version control system. For users, the specific instance of `koala` applicable to the organization should be checked out from the locally controlled repository. This repository should be essentially `read-only` with a specific set of people administering the installation. Until the details can be worked out, use the following checkouts:

- Users:

```
svn co svn://svnserver.qznet/koala
```

Access control isn't set up. Please don't commit to this unless absolutely necessary, and even then run it by an administrator first.

- Administrators:

```
svn co svn://svnserver.qznet/koala
```

Access control isn't set up. Commits should generally be limited to instance-specific areas.

- Developers:

```
git clone (something)
```

1.3 Setting up virtualenv

See <http://simononsoftware.com/virtualenv-tutorial-part-2/> for a more detailed explanation.

1. Install `virtualenv` from the standard repository.

```
sudo aptitude install python-virtualenv virtualenvwrapper
```

2. Create a directory for the virtual environments.


```
mkdir ~/.virtualenvs
```

3. Tell virtualenvwrapper where the folder you just created is. Put it into the bashrc so that you don't have to do it every time you restart.

```
echo "export WORKON_HOME=$HOME/.virtualenvs" >> ~/.bashrc
```

Start up a fresh shell.

4. Create a new virtualenv with the correct interpreter version. Don't use system packages.

If pyenv is controlling the python version,

```
cd /path/to/koala/checkout/trunk/
mkvirtualenv -p 'pyenv which python' --no-site-packages koala
```

If you're just using system python,

```
mkvirtualenv --no-site-packages koala
```

5. mkvirtualenv leaves you with the new virtualenv active. To deactivate,

```
deactivate
```

To reactivate the virtualenv, which you should do when running the scripts in a new terminal:

```
workon koala
```

1.4 Installing the Dependencies

1. Install required python libraries (virtualenv should be active):

```
cd /path/to/koala/checkout/trunk/
pip install -r requirements.txt
```

2. Install sofficehelpers:

sofficehelpers is a collection of scripts to deal with libreoffice documents. As of this version, this only includes a small script (ssconvertor) to convert a spreadsheet into csv files. The libreoffice python interface (uno) requires the use of the python bundled into libreoffice, and therefore is kept separate from the rest of koala. There are plenty of other (and simpler) ways to achieve the same effect, including a number of uno-based scripts to do this. The custom script is retained for the moment to maintain a functional base upon which additional functionality can be added on as needed. If another solution is to be used instead, appropriate changes should be made to `utils.libreoffice.XLFile._make_csv_files()` and `utils.libreoffice.XLFile._parse_sscout()`.

- (a) Install dependencies:

```
sudo apt-get install python-uno
```

- (b) Determine libreoffice python version:

TBD. Until then, refer to “<https://code.google.com/p/slidespeech/wiki/FindingLibreOfficePython>”

- (c) Checkout the officehelpers code:

```
svn co svn://svnserver.qznet/scripts/libreoffice
```

- (a) Navigate to the trunk folder of the obtained repository in a terminal.
- (b) Install to system using:

```
sudo python setup.py install
```

- 4. To be able to generate the documentation, also install the following:

TBD

CONVENTIONS

2.1 gEDA Symbol Conventions

gEDA Symbols contain the primary data used by the scripts. As such, these symbols need to contain the necessary attributes in the correct format.

The conventions listed here are in addition to the standard guidelines and conventions listed on the gEDA wiki at [gEDA/gaf Symbol Creation](#). In the few instances where the conventions listed here conflict with the standard guidelines, the conventions listed here should take precedence.

Some attributes listed here are not standard gEDA/gschem attributes, and as such will cause errors with `gsymcheck`. This is, unfortunately, unavoidable without introducing overly unwieldy annotation schemes.

2.1.1 Symbol Attributes

refdes As per standard gEDA usage.

device This is used as a device class field, with a restricted set of allowed values. See [Device Classes](#).

value The content of this field is Device Class dependent. See [Values](#) for details.

footprint As per standard gEDA usage. This should ideally be a standard footprint name, though at present the scripts don't care.

description (Optional) Description string, usually the title of the datasheet.

symversion As per standard gEDA usage.

status (Optional) Defines the status of the symbol. The absence of the attribute entirely is interpreted as *Active*. See [Symbol Status](#) for more information.

fillstatus (Optional) Defines if a component is soldered or not. The symbol should not contain this attribute. The attribute should be added as needed in the schematic. DNP means Do Not Populate. Absence of the attribute altogether indicate normal usage.

group (Optional) Defines the group of components in the schematic in which the component is included. The symbol should not contain this attribute. The attribute should be added as needed in the schematic. The absence of the attribute entirely is interpreted as the component being in the *default* group. See [Component Groups](#) for an introduction to component groups.

package (Optional) Defines the package of the component. This should be the JEDEC package nomenclature for the component. This is intended to be used for 3D model generation.

2.1.2 Device Classes

The following are the allowed values for the Device Attributes. More should be added as needed.

RES SMD : SMD Resistors.

RES THRU : THRU Resistors.

RES POWER : Off-PCB power resistors for direct mounting onto heatsinks.

RES ARRAY THRU : THRU Resistor Arrays.

RES ARRAY SMD : SMD Resistor Arrays.

POT TRIM : Trimpots.

VARISTOR : Varistors and MOVs.

CAP CER SMD : SMD Ceramic Capacitors.

CAP TANT SMD : SMD Tantalum Capacitors.

CAP CER THRU : THRU Ceramic Capacitors.

CAP ELEC THRU : THRU Electrolytic Capacitors.

CAP POLY THRU : THRU Poly Capacitors.

CAP PAPER THRU : THRU Paper Capacitors.

INDUCTOR SMD : SMD Inductors.

INDUCTOR THRU : THRU Inductors.

FERRITE BEAD SMD : SMD Ferrite Beads.

TRANSFORMER HEAVY : Transformers.

TRANSFORMER SMD : SMD Transformers.

DIODE SMD : SMD Diodes.

DIODE THRU : THRU Diodes.

ZENER SMD : SMD Zener Diodes.

ZENER THRU : THRU Zener Diodes.

TRIAC : Triacs.

LED SMD : SMD LEDs.

LED THRU : THRU LEDs.

LED MODULE : LED Modules.

BRIDGE RECTIFIER : Bridge Rectifiers.

CRYSTAL AT : AT cut Crystals.

CRYSTAL TF : Tuning Fork Crystals.

CRYSTAL OSC : Integrated Crystal Oscillators.

TRANSISTOR THRU : THRU Transistors including MOSFETs.

TRANSISTOR SMD : SMD Transistors including MOSFETs.

IC THRU : THRU Hole ICs.

IC SMD : SMD ICs.

IC PLCC : PLCC ICs, separated because of their need for a socket.

IC POWER : Off-PCB power ICs for direct mounting onto heatsinks.

SOCKET STRIP : SIP sockets.

SOCKET DIP : IC sockets and bases.

RELAY : Relays.

MODULE : Modules.

PCB : Printed Circuit Board.

BUZZER : Buzzers.

CONN BANANA : Banana Connectors.

CONN BERG STRIP : Berg Strips.

CONN TERMINAL BLOCK : Terminal Blocks. Usually single-part.

CONN TERMINAL : Terminal Connectors. Usually two-part.

CONN DTYPE : DTYPE Connectors.

CONN INTERBOARD : Stackthrough Headers.

CONN FRC : FRC Connectors.

CONN MINIDIN : MiniDIN Connectors.

CONN MOLEX : Molex Connector.

CONN MOLEX MINIFIT : Molex Minifit Male (PCB Mount) connectors.

CONN BARREL : DC Power Jacks and similar barrel connectors.

CONN SIP : SIP connectors Male (PCB Mount).

CONN STEREO : Stereo Connectors.

CONN DF13 : Hirose DF13 Connectors.

CONN MODULAR : Modular Connectors.

SWITCH TACT : Tactile Switches.

SWITCH PUSHBUTTON : Pushbutton Switches.

TESTPOINT : Testpoints.

SOLDER DOT : Solder Dots.

2.1.3 Values

General

For the general case, value should include the manufacturer part number. The part number should be the minimal string necessary to uniquely locate components with all parameters of interest, including Grade, Package, etc.

Unless otherwise specified, canonical representation for each class is constructed as `DEVICE VALUE FOOTPRINT`.

Resistors

- Applies to RES SMD, RES THRU, RES POWER, RES ARRAY THRU, RES ARRAY SMD, POT TRIM.
- The value contains the actual resistance value in a standard form.
- Order specifiers to be used are m, E, K, M, G. The Ohm symbol is excluded.
- The numerical part of the value should be greater than 1 (820E instead of 0.82K)
- For special cases, the full manufacturer part number can be used in place of the resistance value.
- Wattage can optionally (preferably) be specified within value, separated from the resistance value with a /.
- Tolerance, Temperature Coefficient, etc. can also be added similarly to Wattage if needed. If so, the conventions should be amended to reflect the correct order as well as code modifications to any relevant [Script Dependencies](#).
- No spaces should be used.

Examples for Resistor Values :

- 10m/1W
- 10E/0.25W
- 10K/1W
- 10M/0.125W
- 10G/0.25W
- 8K2
- 8.2K (preferred)
- PTF561K0000BZEB

Capacitors

- Applies to CAP CER SMD, CAP TANT SMD, CAP CER THRU, CAP ELEC THRU, CAP POLY THRU, CAP PAPER THRU.
- The value contains the actual capacitance value in a standard form.
- Order specifiers to be used are p, n, u. The F symbol is included. (pF, nF, uF)
- The numerical part of the value should be greater than 1 (100nF instead of 0.1uF)
- For special cases, the full manufacturer part number can be used in place of the capacitance value.
- Voltage can optionally (preferably) be specified within value, separated from the capacitance value with a /. This voltage is interpreted as the minimum voltage necessary.
- If the Voltage is not specified, the voltage is assumed to be the `stdvoltage` parameter in the generator file, if any.
- For now, the Voltage should be specified to what is to be purchased (and not the minimum required).
- Tolerance, Temperature, etc. can also be added similarly to Voltage if needed. If so, the conventions should be amended to reflect the correct order as well as code modifications to any relevant [Script Dependencies](#).
- No spaces should be used.

Examples for Capacitor Values :

- 100nF/50V

- 10uF/25V
- 2.2uF/10V
- 100nF
- 4700uF/63V

Standard Voltages :

CAP CER SMD 0805 : 100V

CAP TANT SMD TANT B : 25V

CAP TANT SMD TANT D : 25V

Diodes

- Applies to DIODE THRU, DIODE SMD, ZENER THRU, ZENER SMD, LED THRU, LED MODULE, BRIDGE RECTIFIER.
- The value contains the standard part number as far as possible.
- For LEDs, the value contains the Color. The size is determined by the footprint.
- LED Modules and other special LEDs have the necessary details in the value.
- Diodes not derived from standard part numbers should be manually handled in transform and map files.

Examples for Diode Idents :

- DIODE THRU 1N4007 ALF400-120
- DIODE THRU 1N5402 ALF600-200
- LED THRU RED LED3
- DIODE SMD LL4148 1206P
- BRIDGE RECTIFIER MB6S TO269AA
- ZENER SMD AZ23C3V6-7-F SOT23
- DIODE SMD PGB102ST23 SOT23

Inductors

- Applies to INDUCTOR SMD, INDUCTOR THRU.
- Given the complexity of Inductor specifications and sourcing, Inductor values should be full manufacturer part numbers.
- For low-end inductors locally obtained, the value attribute can contain the inductance value.
- Order specifiers to be used are n, u, m, with the *H* symbol included (nH, uH, mH)
- Additional specifications can be added by using /. Spaces should be avoided.
- Further guidelines should be developed if inductors are used often.

Crystals

- Applies to CRYSTAL AT, CRYSTAL TF, CRYSTAL OSC.
- VALUE should contain the frequency of the crystal along with units. No spaces.
- For special cases, VALUE can be the full manufacturer part number.

Examples for Crystal Values:

- 11.0592MHz
- 16MHz
- 32.768KHz

Connectors

- DEVICE contains the connector family name as listed previously.
- VALUE contains the number of contacts, gender, direction (ST/RA), and any other parameters that may exist.
- VALUE can include spaces. However, every symbol for connectors of the same family should have a consistent structure.
- For highly specialized connectors, the VALUE attribute contains the manufacturer part number.
- FOOTPRINT almost always duplicates the information present in DEVICE and VALUE, and is therefore excluded from the ident string.

Constructors for Connector Idents:

- CONN INTERBOARD; ESQ-104-12-G-D
- CONN BERG STRIP; 2x05PIN 2R [ST/RA] [L?]
- CONN BERG STRIP; 10PIN 1R [ST/RA] [L?]
- CONN FRC; 10PIN [PM/CM] [ST/RA] [NL/WL]
- CONN SIP; 10PIN [PM/CM] [ST/RA]
- CONN DTYPE; DB25 [PM/CM/WM] [ST/RA] [M/F]
- CONN MOLEX MINIFIT; 10PIN [1R/2R] [M/F] [ST/RA]
- CONN MOLEX; 04PIN PM RA
- CONN TERMINAL; 02PIN [PM/CM] [ST/RA]
- CONN TERMINAL BLOCK; 02PIN [ST/RA/ANG]
- CONN MINIDIN; 04PIN PM [ST/RA]
- CONN MODULAR; SS-60000-009
- CONN DF13; DF13A-5P-1.25H
- CONN BARREL; 2.1MM PM RA
- CONN STEREO; 6.3MM PM RA
- CONN THC; PCC-SMP-K-R
- CONN USB; B RA PM THRU
- CONN USB; mB RA PM SMD

2.1.4 Component Groups

HM

2.1.5 Symbol Status

Symbol status determines how the symbol is handled by the scripts. The `STATUS` attribute, if any, should be within the symbol and not added to the schematic. Within the schematic, the `STATUS` attribute should be visible or should be removed, depending on what the status is. (Details Follow). `STATUS` is, in some sense, an outer-loop version of gEDA's `symversion` attribute.

Allowed Status values:

Active : If the `STATUS` attribute is `Active` or does not exist, then the scripts treat the symbol as `Active`. This means the component is acceptable for normal use, and someone in the Company knows the details of procurement and usage of the component.

Experimental If the `STATUS` is `Experimental`, this means that the component is being considered for use. However, care should be taken because the symbol and footprint are likely untested, the component's sourcing details may not be finalized, so on.

Deprecated If the `STATUS` is `Deprecated`, this means a decision has been made to completely phase out use of this component. During redesign of any production PCB, the use of any `Deprecated` components should be looked at and removed if possible. Converting a `Deprecated` component back into `Active` use should involve a specific discussion of the relative merits. Under no circumstances should an `Obsolete` component be `Active`.

Generator The `STATUS` of `Generator` is a special case, indicating that the component represented by the symbol is not necessarily a real component. If such a symbol has a `VALUE` attribute, then the `VALUE` is the default value for the component and should be valid. The `VALUE` attribute of the symbol should be promoted to the schematic and set appropriately (or created if it does not exist in the symbol). Once the `VALUE` attribute is set, the `STATUS=Generator` attribute should be removed from the component in the schematic. The `VALUE` attribute of any symbol whose `STATUS` is not `Generator` should never be promoted / edited in the schematic under any circumstances.

2.1.6 Attribute Promotion

HM

2.1.7 Script Dependencies

At present, the scripts only depend on a subset of the full allowed range of attribute strings. For the sake of consistency, quality control, and painless additions of features to the scripts, the strings should follow the guidelines listed in this document. The actual requirements are listed here for information and to assist in a gradual migration plan.

`conventions.electronics`

Most of the strings listed here are defined in this module, along with string dependent functions.

`conventions.electronics.eln_ident_transform()` :

If the device string starts with any of the following, it's ident constructor leaves out the footprint.

- `CONN`

- MODULE
- CRYSTAL 4PIN

sourcing.electronics

IC If the device string begins with IC, the value is assumed to be a reasonably complete Manufacturer Part Number.

sourcing.digikey

Description

`sourcing.digikey._search_preprocess()` :

Description

`sourcing.digikey._get_device_catstrings()` :

Description

`sourcing.digikey._tf_resistance_to_canonical()` :

Description

Part II

Developer Documentation

GEDA INTERFACE MODULE DOCUMENTATION

3.1 gEDA ConfigsFile module documentation (`gedaif.conffile`)

```
class gedaif.conffile.ConfigsFile (projectfolder)
    Bases: object
    static get_configs_file (projectfolder)
```

3.2 gEDA Project File module documentation (`gedaif.projfile`)

```
class gedaif.projfile.GedaProjectFile (projectfolder)
    Bases: object
    static strip_line (line)
```

3.3 gEDA BOM Parser module documentation (`gedaif.bomparser`)

```
class gedaif.bomparser.BomLine (line, columns)
    Bases: object

class gedaif.bomparser.GedaBomParser (projectfolder, backend)
    Bases: object
    generate_temp_bom (backend)
    prep_temp_bom ()
    delete_temp_bom ()
    get_lines ()
```

3.4 gEDA gschem module documentation (`gedaif.gschem`)

```
gedaif.gschem.conv_gsch2pdf (schpath)
```

3.5 gEDA gsymbolib Module documentation (`gedaif.gsymbolib`)

```
class gedaif.gsymbolib.GedaSymbol(fpath)
    Bases: object
        _acq_sym(fpath)
        ident
        sym_ok
        is_generator
        is_virtual
class gedaif.gsymbolib.GSymGeneratorFile(sympath)
    Bases: object
        _get_data()
        values
gedaif.gsymbolib.gen_symlib()
gedaif.gsymbolib._jinja_init()
gedaif.gsymbolib.seed_generators()
gedaif.gsymbolib.export_symlib()
```

ENTITY HUB MODULES DOCUMENTATION

4.1 EntityHub Transforms Module documentation (`entityhub.transforms`)

```
class entityhub.transforms.TransformFile (tfpath)
    Bases: object
    get_canonical_repr (contextual)
    get_contextual_repr (canonical)
```

4.2 EntityHub Maps Module documentation (`entityhub.maps`)

```
class entityhub.maps.MapFile (mappath)
    Bases: object
    get_idents ()
    get_upartnos (canonical)
    get_apartnos (canonical)
    get_all_partnos (canonical)
    get_partnos (canonical)
    get_strategy (canonical)
    get_canonical (partno)
    get_user_map ()
```


BOMS MODULE DOCUMENTATION

5.1 Electronic Boms Module documentation (`boms.electronics`)

This module contains various classes for handling structured data present in electronics BOMs. The module is built for use with gEDA with specific additions and/or constraints to the standard gEDA project structure. Use with other EDA tools should be possible by replacing `gedaif` with an EDA tool specific package.

5.1.1 Module Summary:

<code>EntityBase()</code>	Placeholder class for potentially track-able objects.
<code>EntityElnComp([item])</code>	Object containing a single electronic component.
<code>EntityGroup</code>	
<code>import_pcb(cardfolder)</code>	Import PCB and return a populated EntityBom

5.1.2 Module Members:

class `boms.electronics.EntityElnComp` (*item=None*)

Bases: `boms.entitybase.EntityBase`

Object containing a single electronic component.

Accept a `gedaif.bomparser.BomLine` generated through gnetlist's bom backend. The `attrs` file should atleast contain:

- device
- value
- footprint
- fillstatus

Parameters *item* (`gedaif.bomparser.BomLine`) – BomLine containing details of component to be created

define (*refdes, device, value, footprint='', fillstatus=''*)

Define the component.

Can be used directly for special cases when there is no `gedaif.bomparser.BomLine` to pass to the class `__init__` function.

Parameters

- **refdes** – Refdes string.
- **device** – Device string.
- **value** – Value string.
- **footprint** – Footprint string. Optional.
- **fillstatus** – Fillstatus string. Optional.

defined

State of the component. The component should be used only when it is fully defined.

This is a read-only property.

refdes

Refdes string.

fillstatus

Fillstatus string. When `fillstatus` is DNP, the component is not included in BOMs irrespective of other configuration states.

ident**device**

Component device string.

value

Component value string.

footprint

Component footprint string. MY- at the beginning of the footprint string is stripped away automatically.

class `boms.electronics.EntityElnGroup` (*groupname*)

Bases: `boms.entitybase.EntityGroupBase`

Container for a group of `EntityElnComp` objects.

Variables

- **groupname** – Name of the group
- **eln_comp_list** – List of `EntityElnComp` objects

insert (*item*)

Insert an electronic component into the `EntityGroup`.

Accept a `BomLine` and generate an `EntityElnComp` to represent the component if it's `fillstatus` is not DNP. Insert the created `EntityElnComp` object into the group.

Parameters *item* (*[gedaif.bomparser.BomLine](#)*) – `BomLine` representing the item to insert.

insert_comp (*comp*)

Insert a manually created component into the `EntityGroup`.

This should be used for components not originating directly from gEDA's gnetlist output.

Parameters *comp* (*`EntityElnComp`*) – Existing component to insert

class `boms.electronics.EntityElnBomConf` (*configdata*)

Bases: `object`

get_configurations ()

get_configsections ()

get_sec_groups (*sectionname*, *config*)

```

    get_configuration (configname)

class boms.electronics.EntityElnBom (configfile)
    Bases: boms.entitybase.EntityBomBase

    create_groups ()

    find_tgroup (item)
        :rtype : EntityElnGroup

    find_group (groupname)
        :rtype : EntityElnGroup

    populate_bom ()

    create_output_bom (configname)

class boms.electronics.OutputBomDescriptor (pcbname, cardfolder, configname, configurations,
                                           multiplier=1)
    Bases: object

class boms.electronics.OutputBomLine (comp, parent)
    Bases: object

    add (comp)

    quantity

class boms.electronics.OutputBom (descriptor)
    Bases: object

    sort_by_ident ()

    find_by_ident (ident)

    insert_component (item)

    multiply (factor, composite=False)

class boms.electronics.CompositeOutputBomLine (line, colcount)
    Bases: object

    add (line, column)

    quantity

class boms.electronics.CompositeOutputBom (bom_list)

    insert_bom (bom, i)

    insert_line (line, i)

```

find_by_ident (*line*)

sort_by_ident ()

`boms.electronics.import_pcb(cardfolder)`

Import PCB and return a populated EntityBom

Accept cardfolder as an argument and return a populated EntityBOM. The cardfolder should be the path to a PCB folder, containing the file structure described in *somewhere*.

See also:

- `gedaif.projfile.GedaProjectFile`
- gEDA Project Folder Structure

Parameters `cardfolder` (*str*) – PCB folder (containing schematic, pcb, gerber)

Returns Populated EntityBom

Return type EntityBom

Example

```
>>> import boms.electronics
>>> bom = boms.electronics.import_pcb('path/to/cardfolder')
```

INVENTORY MODULE DOCUMENTATION

6.1 Electronics Inventory module documentation (`inventory.electronics`)

6.2 Inventory Acquire Module documentation (`inventory.acquire`)

class `inventory.acquire.StockXlsReader` (*xf, sname, location, tfpath*)
Bases: `object`

`_skip_to_header()`

`_get_cols` (*row*)

static `_is_balance` (*item*)

`_row_gen()`

`_tf_row_gen()`

`inventory.acquire.get_stockxlsreader` (*xlpath, sname, location, tfpath*)

`inventory.acquire.get_reader` (*elec_inven_data_idx*)

`inventory.acquire.gen_canonical_transform` (*elec_inven_data_idx*)

`inventory.acquire.helper_transform` (*ident*)

SOURCING MODULE DOCUMENTATION

7.1 Electronics Sourcing module documentation (sourcing.electronics)

`sourcing.electronics.gen_vendor_mapfile (vendor_obj)`

`sourcing.electronics.init_vendors ()`

`sourcing.electronics.export_vendor_map_audit (vendor_obj)`

7.2 Vendors module documentation (sourcing.vendors)

class `sourcing.vendors.VendorBase` (*name, dname, pclass, mappath=None, currency_code='INR',
currency_symbol='INR '*)

Bases: object

name

pclass

mappath

map

currency

search_vpnos (*ident*)

get_vpart (*vpartno, ident=None*)

class `sourcing.vendors.VendorPrice` (*moq, price, currency_def*)

Bases: object

moq

native_value

class `sourcing.vendors.VendorPartBase` (*ident, vendor*)

Bases: object

add_price (*price*)

vpno

```
    qtyavail
    manufacturer
    mpartno
    vpartdesc
    ident
    abs_moq
class sourcing.vendors.VendorElnPartBase (ident, vendor)
    Bases: sourcing.vendors.VendorPartBase
    package
    datasheet
```

7.3 Digi-Key Sourcing Module documentation (sourcing.digikey)

```
class sourcing.digikey.VendorDigiKey (name, dname, pclass, mappath=None, cur-
                                     rency_code=None, currency_symbol=None)
    Bases: sourcing.vendors.VendorBase
    get_vpart (vpartno, ident=None)
    search_vpnos (ident)
    static _search_preprocess (device, value, footprint)
    static _process_product_page (soup)
    static _get_device_catstrings (device)
    _process_index_page (soup, device)
    static _get_resultpage_row_pno (row)
    static _get_resultpage_row_unitp (row)
    static _get_resultpage_row_package (row)
    static _get_resultpage_row_minqty (row)
    static _get_resultpage_row_mfgpno (row)
    _process_resultpage_row (row)
    _get_resultpage_parts (soup)
    static _find_exact_match_package (parts, value)
    static _find_consensus_package (parts)
    static _filter_results_unfiltered (parts)
    static _filter_results_byfootprint (parts, footprint)
    static _filter_results_bycpackage (parts, cpackage, strategy)
    _filter_results (parts, value, footprint)
    _process_results_page (soup, value, footprint)
    _get_search_vpnos (device, value, footprint)
```



```

static _tf_resistance_to_canonical (rstr)
static _tf_capacitance_to_canonical (cstr)
static _tf_package_tant_smd (footprint)
_get_searchpage_filters (soup)
static _get_default_urlparams ()
static _get_searchurl_res_smd ()
static _get_searchurl_cap_cer_smd ()
static _get_searchurl_cap_tant_smd ()
_get_searchurl_filters (searchurl)
_get_pas_vpnos (device, value, footprint)
class sourcing.digikey.DigiKeyElnPart (dkpartno, ident=None, vendor=None)
Bases: sourcing.vendors.VendorElnPartBase

_get_data ()
_get_prices (soup)
static _get_mpartno (soup)
static _get_manufacturer (soup)
static _get_package (soup)
static _get_datasheet_link (soup)
static _get_avail_qty (soup)
static _get_description (soup)
get_price (qty)

```

7.4 CSIL Sourcing Module documentation (sourcing.csil)

```

sourcing.csil.get_csil_prices (params={'layers': 2, 'finish': 'Au', 'pcbname': 'Test', 'dY':
    '46.7', 'qty': [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
    16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33,
    34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51,
    52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69,
    70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87,
    88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99], 'dX': '98.2', 'time':
    15})
class sourcing.csil.VendorCSIL (name, dname, pclass)
Bases: sourcing.vendors.VendorBase
search_vpnos (ident)
get_vpart (vpartno, ident=None)

```


DOX MODULE DOCUMENTATION

8.1 Production Dox module documentation (`dox.production`)

`dox.production.gen_pcb_am` (*projfolder*, *configname*, *outpath=None*, *sno=None*)

8.2 gEDA Project Dox module documentation (`dox.gedaproject`)

`dox.gedaproject.gen_confbom` (*projfolder*, *configname*)

`dox.gedaproject.gen_configdoc` (*projfolder*)

`dox.gedaproject.gen_schpdf` (*projfolder*)

`dox.gedaproject.gen_masterdoc` (*projfolder*)

`dox.gedaproject.gen_confpdf` (*projfolder*, *configname*)

8.3 Dox Render module documentation (`dox.render`)

`dox.render.jinja2_pdfinit` ()

`dox.render.render_pdf` (*stage*, *template*, *outpath*)

UTILS MODULE DOCUMENTATION

9.1 Config Module Documentation (`utils.config`)

9.2 Currency module documentation (`utils.currency`)

class `utils.currency.CurrencyDefinition` (*code, symbol=None, exchval=None*)

Bases: `object`

symbol

exchval

static `_get_exchval` (*code*)

class `utils.currency.CurrencyValue` (*val, currency_def*)

Bases: `object`

native_value

native_string

source_value

source_string

source_currency

9.3 Filesystem Utils Module Documentation (`utils.fsutils`)

class `utils.fs.VersionedOutputFile` (*pathname, numsavedversions=3*)

Create a new output file.

‘*pathname*’ is the name of the file to [over]write. ‘*numSavedVersions*’ tells how many of the most recent versions of ‘*pathname*’ to save.

close ()

as_file ()

Return self’s shadowed file object, since marshal is pretty insistent on working w. pure file objects.

_replace_current_file ()

Replace the current contents of self’s named file.

_backup_current_file ()

Save a numbered backup of self’s named file.

```
    _versioned_name (revision)
        Get self's pathname with a revision number appended.

    _current_revision ()
        Get the revision number of self's largest existing backup.

    _revisions ()
        Get the revision numbers of all of self's backups.

    _delete_old_revisions ()
        Delete old versions of self's file, so that at most self._numSavedVersions versions are retained.

utils.fs.fsutils_cleanup()
```

9.4 LibreOffice Utils Module Documentation (`utils.ooutils`)

```
class utils.libreoffice.XLFile (filepath)
    Bases: object

    get_csv_path (sheetname)

    _make_copy ()

    _make_csv_files ()

    _parse_sscout (string)

    close ()

    _clean ()

utils.libreoffice.get_xlf (fpath)

utils.libreoffice.ooutils_cleanup()
```

9.5 PDF Utils Module documentation (`utils.pdfutils`)

This module contains small helper functions to handle PDF files. These functions are independent of the rest of the software and can be reused wherever needed.

```
utils.pdf.merge_pdf (pdflist, outfilepath)
```

Merges PDF files. Sources are PDF files whose paths are listed in `pdflist` and the combined PDF is written out to `outfilepath`.

Parameters

- **pdflist** – List of paths to PDF files to be merged.
- **outfilepath** – Path where output PDF file is to be written.

Returns `outfilepath`

```
utils.pdf.conv_ps2pdf (pspath, pdfpath)
```

Runs *ps2pdf* to convert a PS file specified in `pspath` to a PDF file written out to `pdfpath`.

Parameters

- **pspath** – Path to PS file to be converted.
- **pdfpath** – Path of PDF file to be generated.

9.6 WWW Utils Module Documentation (`utils.wwutils`)

```
utils.ww.strencode (string)
utils.ww.dump_redirect_cache ()
class utils.ww.CachingRedirectHandler
    Bases: urllib2.HTTPRedirectHandler
        http_error_301 (req, fp, code, msg, headers)
        http_error_302 (req, fp, code, msg, headers)
utils.ww._test_opener (openr)
utils.ww._create_opener ()
utils.ww.urlopen (url)
utils.ww.get_soup (url)
```


b

`boms.electronics`, 21

d

`dox.gedaproject`, 31

`dox.production`, 31

`dox.render`, 31

e

`entityhub.maps`, 19

`entityhub.transforms`, 19

g

`gedaif.bomparser`, 17

`gedaif.conf file`, 17

`gedaif.gschem`, 17

`gedaif.gsymlib`, 17

`gedaif.projfile`, 17

i

`inventory.acquire`, 25

`inventory.electronics`, 25

s

`sourcing.csil`, 29

`sourcing.digikey`, 28

`sourcing.electronics`, 27

`sourcing.vendors`, 27

u

`utils.config`, 33

`utils.currency`, 33

`utils.fs`, 33

`utils.libreoffice`, 34

`utils.pdf`, 34

`utils.www`, 34

Symbols

- `_acq_sym()` (gedaif.gsymlib.GedaSymbol method), 18
- `_backup_current_file()` (utils.fs.VersionedOutputFile method), 33
- `_clean()` (utils.libreoffice.XLFile method), 34
- `_create_opener()` (in module utils.www), 35
- `_current_revision()` (utils.fs.VersionedOutputFile method), 34
- `_delete_old_revisions()` (utils.fs.VersionedOutputFile method), 34
- `_filter_results()` (sourcing.digikey.VendorDigiKey method), 28
- `_filter_results_bycpackage()` (sourcing.digikey.VendorDigiKey static method), 28
- `_filter_results_byfootprint()` (sourcing.digikey.VendorDigiKey static method), 28
- `_filter_results_unfiltered()` (sourcing.digikey.VendorDigiKey static method), 28
- `_find_consensus_package()` (sourcing.digikey.VendorDigiKey static method), 28
- `_find_exact_match_package()` (sourcing.digikey.VendorDigiKey static method), 28
- `_get_avail_qty()` (sourcing.digikey.DigiKeyElnPart static method), 29
- `_get_cols()` (inventory.acquire.StockXlsReader method), 25
- `_get_data()` (gedaif.gsymlib.GSymGeneratorFile method), 18
- `_get_data()` (sourcing.digikey.DigiKeyElnPart method), 29
- `_get_datasheet_link()` (sourcing.digikey.DigiKeyElnPart static method), 29
- `_get_default_urlparams()` (sourcing.digikey.VendorDigiKey static method), 29
- `_get_description()` (sourcing.digikey.DigiKeyElnPart static method), 29
- `_get_device_catstrings()` (sourcing.digikey.VendorDigiKey static method), 28
- `_get_exchval()` (utils.currency.CurrencyDefinition static method), 33
- `_get_manufacturer()` (sourcing.digikey.DigiKeyElnPart static method), 29
- `_get_mpartno()` (sourcing.digikey.DigiKeyElnPart static method), 29
- `_get_package()` (sourcing.digikey.DigiKeyElnPart static method), 29
- `_get_pas_vpnos()` (sourcing.digikey.VendorDigiKey method), 29
- `_get_prices()` (sourcing.digikey.DigiKeyElnPart method), 29
- `_get_resultpage_parts()` (sourcing.digikey.VendorDigiKey method), 28
- `_get_resultpage_row_mfgpno()` (sourcing.digikey.VendorDigiKey static method), 28
- `_get_resultpage_row_minqty()` (sourcing.digikey.VendorDigiKey static method), 28
- `_get_resultpage_row_package()` (sourcing.digikey.VendorDigiKey static method), 28
- `_get_resultpage_row_pno()` (sourcing.digikey.VendorDigiKey static method), 28
- `_get_resultpage_row_unitp()` (sourcing.digikey.VendorDigiKey static method), 28
- `_get_search_vpnos()` (sourcing.digikey.VendorDigiKey method), 28
- `_get_searchpage_filters()` (sourcing.digikey.VendorDigiKey method), 29
- `_get_searchurl_cap_cer_smd()` (sourcing.digikey.VendorDigiKey static method), 29
- `_get_searchurl_cap_tant_smd()` (sourcing.digikey.VendorDigiKey static method), 29

`_get_searchurl_filters()` (sourcing.digikey.VendorDigiKey method), 29

`_get_searchurl_res_smd()` (sourcing.digikey.VendorDigiKey static method), 29

`_is_balance()` (inventory.acquire.StockXlsReader static method), 25

`_jinja_init()` (in module gedaif.gsymlib), 18

`_make_copy()` (utils.libreoffice.XLFile method), 34

`_make_csv_files()` (utils.libreoffice.XLFile method), 34

`_parse_sscout()` (utils.libreoffice.XLFile method), 34

`_process_index_page()` (sourcing.digikey.VendorDigiKey method), 28

`_process_product_page()` (sourcing.digikey.VendorDigiKey static method), 28

`_process_resultpage_row()` (sourcing.digikey.VendorDigiKey method), 28

`_process_results_page()` (sourcing.digikey.VendorDigiKey method), 28

`_replace_current_file()` (utils.fs.VersionedOutputFile method), 33

`_revisions()` (utils.fs.VersionedOutputFile method), 34

`_row_gen()` (inventory.acquire.StockXlsReader method), 25

`_search_preprocess()` (sourcing.digikey.VendorDigiKey static method), 28

`_skip_to_header()` (inventory.acquire.StockXlsReader method), 25

`_test_opener()` (in module utils.www), 35

`_tf_capacitance_to_canonical()` (sourcing.digikey.VendorDigiKey static method), 29

`_tf_package_tant_smd()` (sourcing.digikey.VendorDigiKey static method), 29

`_tf_resistance_to_canonical()` (sourcing.digikey.VendorDigiKey static method), 28

`_tf_row_gen()` (inventory.acquire.StockXlsReader method), 25

`_versioned_name()` (utils.fs.VersionedOutputFile method), 33

A

`abs_moq` (sourcing.vendors.VendorPartBase attribute), 28

`add()` (boms.electronics.CompositeOutputBomLine method), 23

`add()` (boms.electronics.OutputBomLine method), 23

`add_price()` (sourcing.vendors.VendorPartBase method), 27

`as_file()` (utils.fs.VersionedOutputFile method), 33

B

`BomLine` (class in gedaif.bomparser), 17

`boms.electronics` (module), 21

C

`CachingRedirectHandler` (class in utils.www), 35

`close()` (utils.fs.VersionedOutputFile method), 33

`close()` (utils.libreoffice.XLFile method), 34

`CompositeOutputBom` (class in boms.electronics), 23

`CompositeOutputBomLine` (class in boms.electronics), 23

`ConfigsFile` (class in gedaif.conf), 17

`conv_gsch2pdf()` (in module gedaif.gsch), 17

`conv_ps2pdf()` (in module utils.pdf), 34

`create_groups()` (boms.electronics.EntityElnBom method), 23

`create_output_bom()` (boms.electronics.EntityElnBom method), 23

`currency` (sourcing.vendors.VendorBase attribute), 27

`CurrencyDefinition` (class in utils.currency), 33

`CurrencyValue` (class in utils.currency), 33

D

`datasheet` (sourcing.vendors.VendorElnPartBase attribute), 28

`define()` (boms.electronics.EntityElnComp method), 21

`defined` (boms.electronics.EntityElnComp attribute), 22

`delete_temp_bom()` (gedaif.bomparser.GedaBomParser method), 17

`device` (boms.electronics.EntityElnComp attribute), 22

`DigiKeyElnPart` (class in sourcing.digikey), 29

`dox.gedaproject` (module), 31

`dox.production` (module), 31

`dox.render` (module), 31

`dump_redirect_cache()` (in module utils.www), 35

E

`EntityElnBom` (class in boms.electronics), 23

`EntityElnBomConf` (class in boms.electronics), 22

`EntityElnComp` (class in boms.electronics), 21

`EntityElnGroup` (class in boms.electronics), 22

`entityhub.maps` (module), 19

`entityhub.transforms` (module), 19

`exchval` (utils.currency.CurrencyDefinition attribute), 33

`export_symlib()` (in module gedaif.gsymlib), 18

`export_vendor_map_audit()` (in module sourcing.electronics), 27

F

`fillstatus` (boms.electronics.EntityElnComp attribute), 22

`find_by_idnt()` (boms.electronics.CompositeOutputBom method), 23

find_by_ident() (boms.electronics.OutputBom method), 23
 find_group() (boms.electronics.EntityElnBom method), 23
 find_tgroup() (boms.electronics.EntityElnBom method), 23
 footprint (boms.electronics.EntityElnComp attribute), 22
 fsutils_cleanup() (in module utils.fs), 34

G

GedaBomParser (class in gedaif.bomparser), 17
 gedaif.bomparser (module), 17
 gedaif.conf file (module), 17
 gedaif.gschem (module), 17
 gedaif.gsymlib (module), 17
 gedaif.projfile (module), 17
 GedaProjectFile (class in gedaif.projfile), 17
 GedaSymbol (class in gedaif.gsymlib), 18
 gen_canonical_transform() (in module inventory.acquire), 25
 gen_confbom() (in module dox.gedaproject), 31
 gen_configdoc() (in module dox.gedaproject), 31
 gen_confpdf() (in module dox.gedaproject), 31
 gen_masterdoc() (in module dox.gedaproject), 31
 gen_pcb_am() (in module dox.production), 31
 gen_schpdf() (in module dox.gedaproject), 31
 gen_symlib() (in module gedaif.gsymlib), 18
 gen_vendor_mapfile() (in module sourcing.electronics), 27
 generate_temp_bom() (gedaif.bomparser.GedaBomParser method), 17
 get_all_partnos() (entityhub.maps.MapFile method), 19
 get_apartnos() (entityhub.maps.MapFile method), 19
 get_canonical() (entityhub.maps.MapFile method), 19
 get_canonical_repr() (entityhub.transforms.TransformFile method), 19
 get_configs_file() (gedaif.conf file.ConfigsFile static method), 17
 get_configsections() (boms.electronics.EntityElnBomConf method), 22
 get_configuration() (boms.electronics.EntityElnBomConf method), 22
 get_configurations() (boms.electronics.EntityElnBomConf method), 22
 get_contextual_repr() (entityhub.transforms.TransformFile method), 19
 get_csil_prices() (in module sourcing.csil), 29
 get_csv_path() (utils.libreoffice.XLFile method), 34
 get_ids() (entityhub.maps.MapFile method), 19
 get_lines() (gedaif.bomparser.GedaBomParser method), 17
 get_partnos() (entityhub.maps.MapFile method), 19
 get_price() (sourcing.digikey.DigiKeyElnPart method), 29

get_reader() (in module inventory.acquire), 25
 get_sec_groups() (boms.electronics.EntityElnBomConf method), 22
 get_soup() (in module utils.www), 35
 get_stockxlsreader() (in module inventory.acquire), 25
 get_strategy() (entityhub.maps.MapFile method), 19
 get_upartnos() (entityhub.maps.MapFile method), 19
 get_user_map() (entityhub.maps.MapFile method), 19
 get_vpart() (sourcing.csil.VendorCSIL method), 29
 get_vpart() (sourcing.digikey.VendorDigiKey method), 28
 get_vpart() (sourcing.vendors.VendorBase method), 27
 get_xlfile() (in module utils.libreoffice), 34
 GSymGeneratorFile (class in gedaif.gsymlib), 18

H

helper_transform() (in module inventory.acquire), 25
 http_error_301() (utils.www.CachingRedirectHandler method), 35
 http_error_302() (utils.www.CachingRedirectHandler method), 35

I

ident (boms.electronics.EntityElnComp attribute), 22
 ident (gedaif.gsymlib.GedaSymbol attribute), 18
 ident (sourcing.vendors.VendorPartBase attribute), 28
 import_pcb() (in module boms.electronics), 24
 init_vendors() (in module sourcing.electronics), 27
 insert() (boms.electronics.EntityElnGroup method), 22
 insert_bom() (boms.electronics.CompositeOutputBom method), 23
 insert_comp() (boms.electronics.EntityElnGroup method), 22
 insert_component() (boms.electronics.OutputBom method), 23
 insert_line() (boms.electronics.CompositeOutputBom method), 23
 inventory.acquire (module), 25
 inventory.electronics (module), 25
 is_generator (gedaif.gsymlib.GedaSymbol attribute), 18
 is_virtual (gedaif.gsymlib.GedaSymbol attribute), 18

J

jinja2_pdfinit() (in module dox.render), 31

M

manufacturer (sourcing.vendors.VendorPartBase attribute), 28
 map (sourcing.vendors.VendorBase attribute), 27
 MapFile (class in entityhub.maps), 19
 mappath (sourcing.vendors.VendorBase attribute), 27
 merge_pdf() (in module utils.pdf), 34
 moq (sourcing.vendors.VendorPrice attribute), 27

mpartno (sourcing.vendors.VendorPartBase attribute), 28
multiply() (boms.electronics.OutputBom method), 23

N

name (sourcing.vendors.VendorBase attribute), 27
native_string (utils.currency.CurrencyValue attribute), 33
native_value (sourcing.vendors.VendorPrice attribute), 27
native_value (utils.currency.CurrencyValue attribute), 33

O

ooutils_cleanup() (in module utils.libreoffice), 34
OutputBom (class in boms.electronics), 23
OutputBomDescriptor (class in boms.electronics), 23
OutputBomLine (class in boms.electronics), 23

P

package (sourcing.vendors.VendorElnPartBase attribute), 28
pclass (sourcing.vendors.VendorBase attribute), 27
populate_bom() (boms.electronics.EntityElnBom method), 23
prep_temp_bom() (gedaif.bomparser.GedaBomParser method), 17

Q

quantity (boms.electronics.CompositeOutputBomLine attribute), 23
quantity (boms.electronics.OutputBomLine attribute), 23

R

refdes (boms.electronics.EntityElnComp attribute), 22
render_pdf() (in module dox.render), 31

S

search_vpnos() (sourcing.csil.VendorCSIL method), 29
search_vpnos() (sourcing.digikey.VendorDigiKey method), 28
search_vpnos() (sourcing.vendors.VendorBase method), 27
seed_generators() (in module gedaif.gsymlib), 18
sort_by_ident() (boms.electronics.CompositeOutputBom method), 24
sort_by_ident() (boms.electronics.OutputBom method), 23
source_currency (utils.currency.CurrencyValue attribute), 33
source_string (utils.currency.CurrencyValue attribute), 33
source_value (utils.currency.CurrencyValue attribute), 33
sourcing.csil (module), 29
sourcing.digikey (module), 28
sourcing.electronics (module), 27
sourcing.vendors (module), 27
StockXlsReader (class in inventory.acquire), 25

strencode() (in module utils.www), 35
strip_line() (gedaif.projfile.GedaProjectFile static method), 17
sym_ok (gedaif.gsymlib.GedaSymbol attribute), 18
symbol (utils.currency.CurrencyDefinition attribute), 33

T

TransformFile (class in entityhub.transforms), 19

U

urlopen() (in module utils.www), 35
utils.config (module), 33
utils.currency (module), 33
utils.fs (module), 33
utils.libreoffice (module), 34
utils.pdf (module), 34
utils.www (module), 34

V

value (boms.electronics.EntityElnComp attribute), 22
values (gedaif.gsymlib.GSymGeneratorFile attribute), 18
VendorBase (class in sourcing.vendors), 27
VendorCSIL (class in sourcing.csil), 29
VendorDigiKey (class in sourcing.digikey), 28
VendorElnPartBase (class in sourcing.vendors), 28
VendorPartBase (class in sourcing.vendors), 27
VendorPrice (class in sourcing.vendors), 27
VersionedOutputFile (class in utils.fs), 33
vpardesc (sourcing.vendors.VendorPartBase attribute), 28
vpno (sourcing.vendors.VendorPartBase attribute), 27
vqtyavail (sourcing.vendors.VendorPartBase attribute), 27

X

XLFile (class in utils.libreoffice), 34