

Blunk electronic
owner: Dipl.-Ing. Mario Blunk
Buchfinkenweg 3
99097 Erfurt / Germany

cellular	+49 (0) 176 2904 5855
land line	+49 (0) 361 6022 5184
email	mario.blunk@blunk-electronic.de
web	www.blunk-electronic.de



User and Installation Guide for Stock Managing Software

Stock Manager (ERP)

Doc. Version 2015-11-20-1

Abstract: User and installation guide for Stock Manager on Linux operating systems. Audience are electronic manufacturing services providing companies who want to migrate to Linux based resource management. Outstanding features: Stock Manager does not tie the user to a particular software manufacturer, is Open Source and mission critical programmed in Ada.

Keywords: manufacturer, distributor, part code, order code, URL datasheet, storage place, reserved, quantity, net price, BOM, EAGLE, CAE, facility, order list, withdrawal list, status production, obsolete, active, discontinued, not for new designs, checkout, query, data base, csv, LibreOffice, Linux, Command Line, CLI, graphical user interface, GUI, Ada, GNAT, import, compiler, plain text, log, ssh, server, NFS, Samba, SMB, installation support, GNU, make, SQL, git

1 Introduction

As a company that provides electronic manufacturing services (EMS) you need a stock managing system in order not to get lost among hundreds of card board boxes and plastic bags. Stock managing systems of various kinds are available on the market but what they frequently lack is easy to point out:

- ➔ portability/readability regardless of software manufacturer/vendor
- ➔ manual/human interaction if stock managing software fails
- ➔ open sourcing the code in order to allow extension, modification, bug fixing
- ➔ Linux operating system support

If these requirements are met, you can focus on your actual work:

Making Electronic Products !

I appreciate all kinds of critics to improve the quality of this document !

Contents

1 Introduction.....	2
2 Key Features of Stock Manager.....	5
3 The Concept Behind.....	6
4 How to Operate Stock Manager ?.....	7
4.1 Show an Item.....	7
4.1.1 Show by ID.....	7
4.1.2 Show by Facility Code.....	8
4.1.3 Show by Order Code of a Distributor or Manufacturer.....	10
4.1.4 Show Full Information.....	11
4.2 Editing an Item.....	12
4.2.1 Facility Data.....	12
4.2.1.1 Quantity on Stock	12
4.2.1.2 Quantity Reserved	13
4.2.1.3 Storage Place	14
4.2.2 Manufacturer Data.....	15
4.2.2.1 Name.....	15
4.2.2.2 Part Code.....	16
4.2.2.3 URL to Datasheet.....	16
4.2.3 Distributor Data	17
4.2.3.1 Name.....	17
4.2.3.2 Order Code.....	18
4.2.3.3 Quantity Min.....	19
4.2.3.4 Price Net.....	20
4.3 Adding an Item to Stock.....	21
4.4 Deleting an Item from Stock.....	22
4.5 View Log.....	23
4.6 Rollback (UnDo).....	24
4.7 Macro Functions.....	25
4.7.1 Generate Facility BOM.....	26
4.7.2 Querying Items by Facility BOM.....	27
4.7.3 Checking Out Items by Facility BOM.....	28
4.8 Getting Help.....	29
5 Facility Part Code Syntax.....	30
6 Configuration Files.....	31
7 Installation.....	32
7.1 Stand Alone.....	32
7.2 Server.....	32
8 Limitations.....	33
8.1 Manufacturers.....	33
8.2 Distributors.....	33
8.3 Maximum Number of Item Entries in Stock Data Base.....	33
8.4 Loading and Saving Time.....	33
9 Extendability.....	33
10 Frequently Asked Questions.....	34
11 To Do.....	35

12 Links.....	36
13 Disclaimer.....	37

2 Key Features of Stock Manager

To give you a brief summary of things *Stock Manager* can do for your facility, please go on reading here.

***Stock Manager* keeps record of:**

- in and out flow of items (quantity, reserved, availability, ...)
- storage place, project name, ...
- manufacturer data (name, production status, part code, ...)
- distributor data (name, order code, net price, min. quantity, ...)

***Stock Manager* generates:**

- bill of materials (BOM)
- order lists (OL)
- withdrawal lists (WL)

3 The Concept Behind

The product *Stock Manager* consists of a single executable Linux binary file (ELF) that receives its parameters via the command line. Users who are scared of command line interfaces (CLI) are encouraged to continue reading here. I'll take you by the hand, if necessary via phone or remote administration. Users who are familiar with CLI will easily get along with it. Once gotten used to a CLI you will love its efficiency, I promise !

There are also some configuration and help files and most important: the *stock data base*:

The stock data base is basically a so called CSV-file. CSV means Comma-Separated Values. The separator might also be a Semicolon, a Space, a Colon or what ever. This file format has following outstanding properties:

- ➔ It can be imported with EVERY spread sheet software like LibreOffice or MS-Office
- ➔ You can edit it easily, even with a simple text editor
- ➔ You can read it easily, it is plain text
- ➔ Even if *Stock Manager* fails, if *Blunk electronic* ceases to exist, or if you decide not to work any longer with it, you are not lost ! The work you did editing and adding items to the stock data base can be continued with other tools that can read csv !

**So you never get into the claws of a software house
whose software rides
on a cryptic binary coded data base.**

See section 10 Frequently Asked Questions page 34 for more.

Stock Manager itself refers to the core functions that come with the binary file `stock_manager`. There is no graphical user interface (GUI) with fancy windows and widgets. The very fact that the program is command line based, implies extendability in terms of batch/scripting and graphical user interface. Every skilled software programmer can make a GUI for you on top of *Stock Manager* according to your needs and financial resources. See section 10 Frequently Asked Questions page 34 for more.

Please open this link to get full access to all text, csv and binary files supplied:

https://github.com/Blunk-electronic/stock_manager/archive/v013.zip

4 How to Operate Stock Manager ?

In the following we go right into examples instead of abstract theory. We take the role of the person in your corporation or facility that is authorized to work with *Stock Manager*. We also assume the installation of the program has been done already.

The example files used here are free for download to have a look into. They may also serve as starting point for your own facility.

This is the most important file: The stock data base file

https://github.com/Blunk-electronic/stock_manager/blob/master/example_database/stock_db.csv

You may open it with LibreOffice to have a look into it. Every entry in the data base has an unambiguous ID (column 1) and the *facility part code* used in your company (column 2). These two markers allow locating and editing an entry. For example the ID 13 refers to an item with part code C_PAC_S_0805_VAL_100n. The part code should be unique and speak for the part itself. In the physical stock of your facility, the bag or box that contains the part carries a label with this code or/and the part ID. See section 5 page 30 for details. So much for the basics. You may close the file now.

There are some configuration files we'll discuss later in section 6 page 31.

4.1 Show an Item

Lets start with just a simple query to the data base. To inquire for an item you need to know either its ID or its facility part code.

4.1.1 Show by ID

In your terminal window type command: `stock_manager show_by_id 13` as shown below. The output immediately comes up displaying facility part code, date of last editing, quantities, storage place and miscellaneous information.

```
lager@server2:~> stock_manager show_by_id 13
```

```
STOCK MANAGER version 001
```

```
=====
action          : SHOW_BY_ID
part id         : 13
```

```
-----
part id         : 13
part code       : C_PAC_S_0805_VAL_100n
date edited    : 2015-03-18 18:51:12
qty on stock   : 279
qty reserved   : 0
qty available  : 279
storage place  : n/a
project        : n/a
remarks        : vmax=50V
-----
```

```
lager@server2:~>
```

4.1.2 Show by Facility Code

If the exact part code or a fragment is known, the search can be conducted as shown below. In case of more occurrences all of them are output.

Example #1:

```
lager@server2:~> stock_manager show_by_fac_code C_PAC_S_0805_VAL_100n
```

```
STOCK MANAGER version 001
```

```
=====
action          : SHOW_BY_FAC_CODE
part code       : C_PAC_S_0805_VAL_100n
```

```
-----
part id         : 13
part code       : C_PAC_S_0805_VAL_100n
date edited    : 2015-03-18 18:51:12
qty on stock   : 279
qty reserved   : 0
qty available  : 279
storage place  : n/a
project        : n/a
remarks        : vmax=50V
-----
```

```
-----
part id         : 14
part code       : C_PAC_S_0805_VAL_100n_Vmax_100V
date edited    : 2015-03-13 17:10:34
qty on stock   : 0
qty reserved   : 0
qty available  : 0
storage place  : n/a
project        : n/a
remarks        : n/a
-----
```

```
Number of parts found : 2
lager@server2:~>
```


Example #2:

Assume we want to see if there is a resistor with value 422 ohms.

```
lager@server2:~> stock_manager show_by_fac_code _VAL_422

STOCK MANAGER version 001
=====
action          : SHOW_BY_FAC_CODE
part code       : _VAL_422

-----
part id         : 161
part code       : R_PAC_S_0805_VAL_422
date edited    : 2015-03-18 18:59:01
qty on stock   : 8
qty reserved    : 0
qty available   : 8
storage place   : n/a
project        : n/a
remarks        : n/a
-----
lager@server2:~>
```

For correctness, resistor values below 1 kohms should end with letter R. The resistor in this case should end in VAL_422R. Of course, if there was a capacitor with lets say 422 Nanofarad, this would have been output too.

Wildcards may be used as well. The command

```
stock_manager show_by_fac_code *PAC_S_0805*
```

for example, would return all parts with SMD package typ 0805.

4.1.3 Show by Order Code of a Distributor or Manufacturer

If a fragment or the full order code of a distributor is known, the search can be done as shown below. In case of multiple occurrences, all of them will be output. Usage of option *full* is recommended (see section 4.1.4 page 11):

Example #1:

```
lager@server2:~> stock_manager full show_by_order_code 732736

STOCK MANAGER V009
=====
action          : SHOW_BY_ORDER_CODE
order code      : 732736

-----
part id         : 66
part code       : IC_PAC_S_SO6_VAL_74LVC1G18DCK
date edited     : 2015-05-15 10:34:57
qty on stock    : 6
qty reserved    : 0
qty available   : 6
storage place   : S3/8
project         : n/a
remarks         : n/a

-----
manufacturer 1  : TI
date edited     : 2015-07-28 08:15:00
part code       : SN74LVC1G18DCKR
url datasheet 1 : http://www.farnell.com/datasheets/1737053.pdf
url datasheet 2 : n/a
status prod.    : ACTIVE

-----
distributor 2   : AX
order code      : 732736
date edited     : 2015-07-28 08:16:08
qty min.        : 1
price net       : 0.11

-----
distributor 3   : FARNELL
order code      : 2334935
date edited     : 2015-04-07 11:15:02
qty min.        : 1
price net       : 0.19

-----
lager@server2:~>
```

If the query is about a certain manufacturer part code, the command could be for example:

```
lager@server2:~> stock_manager full show_by_manu_code LM324*
```

4.1.4 Show Full Information

If the *full* option is applied, more information will be output. For EMS, important information is about manufacturers, distributors, prices and production status. Lets do the part search again with the *full* option given to *stock_manager* :

```
lager@server2:~> stock_manager full show_by_id 13

STOCK MANAGER version 001
=====
action          : SHOW_BY_ID
part id         : 13
-----

part id        : 13
part code      : C_PAC_S_0805_VAL_100n
date edited    : 2015-03-18 18:51:12
qty on stock   : 279
qty reserved   : 0
qty available  : 279
storage place  : n/a
project        : n/a
remarks        : vmax=50V
-----

manufacturer 1 : AVX
date edited    : 2015-03-17 14:32:09
part code      : 08055C104JAT2A
url datasheet 1 : http://www.avx.com/docs/Catalogs/cx7r.pdf
url datasheet 2 : n/a
status prod.   : ACTIVE
-----

distributor 1  : REICHELT
order code     : X7R-G0805 100N
date edited    : 2015-03-17 14:43:21
qty min.       : 1
price net      : 0.04
-----

distributor 3  : FARNELL
order code     : 1740673
date edited    : 2015-03-17 14:45:24
qty min.       : 10
price net      : 0.01
-----

lager@server2:~>
```

At this point you may ask, how many manufacturers or distributors are possible. You may also want to add more properties like *RoHS* or *UL* conformity. See section 8 and 9 for limitations and extendability.

4.2 Editing an Item

4.2.1 Facility Data

4.2.1.1 Quantity on Stock

Most frequently parts are to be put on stock or taken from stock. The easiest way to do so for a particular item is shown below. If a bunch of parts is to be withdrawn from stock, a more efficient method is described in section 4.7.3 on page 28. The number of parts added to stock is always a delta. Means if you add one item type 1, if you withdraw one piece type -1:

```
lager@server2:~> stock_manager edit 13 qty_delta_stock 1
```

```
STOCK MANAGER V009
=====
action          : EDIT
part id         : 13
property to edit: QTY_DELTA_STOCK
part code BEL   : C_PAC_S_0805_VAL_100n
storage place   : S1/1
qty reserved    : 0
qty on stock old: 279
qty stock delta : +1
qty on stock new: 280

ARE YOU SURE ? (y/n) :y

lager@server2:~>
```

NOTE: Editing an item by default always requires operator confirmation !

If option *no_confirmation* is applied, no confirmation is requested. **Use with care !** This option is useful when *stock_manager* is called from inside a bash script.

```
lager@server2:~> stock_manager no_confirmation edit 13 qty_delta_stock 1
```

```
STOCK MANAGER version 001
=====
action          : EDIT
part id         : 13
property to edit: QTY_DELTA_STOCK
part code BEL   : C_PAC_S_0805_VAL_100n
storage place   : S1/1
qty reserved    : 0
qty on stock old: 279
qty stock delta : +1
qty on stock new: 280

lager@server2:~>
```

4.2.1.2 Quantity Reserved

Sometimes a certain amount of parts is to be reserved for special uses (like a priority project). Here you again pass a delta of items to be reserved or un-reserved. Upon execution of this command, the number of available items changes:

```
lager@server2:~> stock_manager edit 13 qty_delta_reserved 70
```

```
STOCK MANAGER version 001
=====
action          : EDIT
part id         : 13
property to edit: QTY_DELTA_RESERVED
part code BEL   : C_PAC_S_0805_VAL_100n
storage place   : S1/1
qty on stock    : 279
qty reserved old: 0
qty rsvd. delta : +70
qty reserved new: 70

ARE YOU SURE ? (y/n) :y

lager@server2:~> stock_manager show_by_id 13
```

```
STOCK MANAGER version 001
=====
action          : SHOW_BY_ID
part id         : 13

-----
part id         : 13
part code       : C_PAC_S_0805_VAL_100n
storage place   : S1/1
date edited     : 2015-03-19 17:20:24
qty on stock    : 279
qty reserved    : 70
qty available   : 209
storage place   : n/a
project         : n/a
remarks         : vmax=50V
-----

lager@server2:~>
```

NOTE: Reserved items can not be withdrawn from stock !

4.2.1.3 Storage Place

The place of storage is to be assigned as shown below. Note, if white-spaces are used, wrap the text in double-quotes:

```
lager@server2:~> stock_manager edit 13 storage_place "Shelf 4, Box 23"
```

```
STOCK MANAGER version 001
=====
action          : EDIT
part id         : 13
property to edit: STORAGE_PLACE
part code BEL   : C_PAC_S_0805_VAL_100n
place old       : S1/1
place new       : Shelf 4, Box 23

ARE YOU SURE ? (y/n) :y

lager@server2:~>
```

See section 11 To Do for more.

4.2.2 Manufacturer Data

4.2.2.1 Name

To assign or change the name of the manufacturer do as shown here. Note, if white-spaces are used, wrap the text in double-quotes:

```
lager@server2:~> stock_manager edit 13 manufacturer_2_name multicom
```

```
STOCK MANAGER version 001
=====
action          : EDIT
part id         : 13
property to edit: MANUFACTURER_2_NAME
part code BEL   : C_PAC_S_0805_VAL_100n
manuf. name old : n/a
manuf. name new : multicom

ARE YOU SURE ? (y/n) :y

lager@server2:~>
```

To **remove** the manufacturer from this item, assign a "not assigned" tag:

```
lager@server2:~> stock_manager edit 13 manufacturer_2_name n/a
```

whereupon this manufacturer will not be shown anymore for this particular item.

4.2.2.2 Part Code

Every manufacturer has a nomenclature of part codes. Note: The manufacturer code is not to be confused with the facility part code. Note, if white-spaces are used, wrap the text in double-quotes. The part code of manufacturer 1 is to be changed as shown here:

```
lager@server2:~> stock_manager edit 13 manufacturer_1_part_code 08055C104JAT2A
```

```
STOCK MANAGER version 001
=====
action          : EDIT
part id         : 13
property to edit: MANUFACTURER_1_PART_CODE
part code BEL   : C_PAC_S_0805_VAL_100n
manuf. name     : AVX
part code old   : NCC1701-E
part code new   : 08055C104JAT2A

ARE YOU SURE ? (y/n) :y

lager@server2:~>
```

4.2.2.3 URL to Datasheet

If the URL to a datasheet is to be assigned, type:

```
stock_manager edit 250 manufacturer_1_datasheet_1 http://www.ti.com/BCW60.pdf
```

Note, if white-spaces are used, wrap the text in double-quotes. See section 8 Limitations for maximum number of datasheet links.

4.2.3 Distributor Data

4.2.3.1 Name

To assign or change the name of a distributor do as shown here. Note, if white-spaces are used, wrap the text in double-quotes:

```
lager@server2:~> stock_manager edit 13 distributor_2_name "AX ELECTRONIC"
```

```
STOCK MANAGER version 001
=====
action          : EDIT
part id         : 13
property to edit: DISTRIBUTOR_2_NAME
part code BEL   : C_PAC_S_0805_VAL_100n
dist. name old  : n/a
dist. name new  : AX ELECTRONIC

ARE YOU SURE ? (y/n) :y

lager@server2:~>
```

As pointed out in section 4.2.2.1 page 15 a distributor is delete if the "not-assigned mark" is assigned. Means, if you assign the name "n/a" to a distributor, it will be removed from this particular item.

4.2.3.2 Order Code

Distributors usually have order codes for parts they sell. So the order code must be assigned to an item as follows. The order code may be a number or a string. Note, if white-spaces are used, wrap the text in double-quotes:

```
lager@server2:~> stock_manager edit 13 distributor_2_order_code 445723
```

```
STOCK MANAGER version 001
=====
action          : EDIT
part id         : 13
property to edit: DISTRIBUTOR_2_ORDER_CODE
part code BEL   : C_PAC_S_0805_VAL_100n
distributor name: AX ELECTRONIC
order code old  : n/a
order code new  : 445723

ARE YOU SURE ? (y/n) :y

lager@server2:~>
```

4.2.3.3 Quantity Min

Frequently, there is a minimum of items to buy from a distributor. Assign the minimum quantity as:

```
lager@server2:~> stock_manager edit 13 distributor_2_qty_min 10
```

```
STOCK MANAGER version 001
=====
action          : EDIT
part id         : 13
property to edit: DISTRIBUTOR_2_QTY_MIN
part code BEL   : C_PAC_S_0805_VAL_100n
distributor name: AX ELECTRONIC
dist. order code: 445723
qty min old     : 1
qty min new     : 10

ARE YOU SURE ? (y/n) :y

lager@server2:~>
```

The default minimum quantity is always 1.

4.2.3.4 Price Net

The net price of one item is to be assigned as shown below. When creating the facility BOM (see section 4.7.1 Generate Facility BOM page 26) these figures are taken into account in order to calculate the total material price of a unit:

```
lager@server2:~> stock_manager edit 13 distributor_2_price_net 0.05
```

```
STOCK MANAGER version 001
=====
action          : EDIT
part id         : 13
property to edit: DISTRIBUTOR_2_PRICE_NET
part code BEL   : C_PAC_S_0805_VAL_100n
distributor name: AX ELECTRONIC
price net old   : 0.00
price net new   : 0.05

ARE YOU SURE ? (y/n) :y

lager@server2:~>
```

There is no emphasis on the currency. If you live in a country that uses EUR, regard these figures as EURO. If you work with Russian Ruble, regard them as Russian Ruble.

See section 11 To Do for more.

4.3 Adding an Item to Stock

So far we worked with the existing stock provided with the example stock_data_base:

https://github.com/Blunk-electronic/stock_manager/blob/master/example_database/stock_db.csv

But items also need to be added. Adding an item is done by specifying the facility part code. As this implementation of *Stock Manager* is made for EMS, the syntax of the facility part code must meet some rules. See section 5 Facility Part Code Syntax page 30. The syntax can be adopted to other kinds of items to store. See section 11 To Do for more.

However, you need some rules for the facility part code. Otherwise you eventually get lost among arbitrarily chosen codes.

Now the example:

```
lager@server2:~> stock_manager add IC_PAC_S_SO8_VAL_AT24C08

STOCK MANAGER version 001
=====
action          : ADD
part code       : IC_PAC_S_SO8_VAL_AT24C08

ARE YOU SURE ? (y/n) :y

Part has been added to data base with part_id 274

lager@server2:~>
```

Upon adding an item, a new ID gets assigned. You are now allowed to edit properties like quantities, manufacturer and distributor data.

4.4 Deleting an Item from Stock

If an item is not subject of storage any more, it should be deleted. As a safety measure, an item can be deleted by its facility part code only.

CAUTION: Upon deleting, all information entered before like quantities, manufacturer and distributor data will be lost !

Do as shown in this example:

```
lager@server2:~> stock_manager delete IC_PAC_S_SO8_VAL_AT24C08
```

```
STOCK MANAGER version 001
=====
action          : DELETE
part_code_fac   : IC_PAC_S_SO8_VAL_AT24C08

WARNING : DELETE PART WITH
        ID       : 274
        PART_CODE : IC_PAC_S_SO8_VAL_AT24C08

ARE YOU SURE ? (y/n) :y

lager@server2:~>
```

4.5 View Log

All edit operations, means changes, are logged. If you need to look back on what has been done, see example below:

```
lager@server2:~> stock_manager log

STOCK MANAGER version 001
=====
action          : LOG
stock data base : /home/luno/verwaltung/lager/stock_db.csv

stock manager activities log
-----
DATE
YYYY:MM:DD HH:MM:SS   | action
-----
.
.
.
2015-03-20 08:19:27   | edit  13  distributor_2_name  AX ELECTRONIC
2015-03-20 08:23:20   | edit  13  distributor_2_order_code  445723
2015-03-20 08:26:30   | edit  13  distributor_2_qty_min  10
2015-03-20 08:30:51   | edit  13  distributor_2_price_net  0.05
2015-03-20 08:38:51   | edit  13  distributor_2_name  n/a
2015-03-20 08:48:47   | add  IC_PAC_S_SO8_VAL_AT24C08
2015-03-20 08:54:54   | delete IC_PAC_S_SO8_VAL_AT24C08
lager@server2:~>
```

4.6 Rollback (UnDo)

After every edit operation, a backup of the previous stock data base is made in the background. Every edit operation can be un-done with a so called Rollback. This rollback just restores the stock data base from the most recent backup.

```
lager@server2:~> stock_manager roll_back
```

```
STOCK MANAGER version 001
```

```
=====
```

```
action          : ROLL_BACK
```

```
stock data base : /home/luno/lager/stock_db.csv
```

```
restore from    : /home/luno/lager/bak/BAK_2015-03-20_09-05-39__stock_db.csv
```

```
ARE YOU SURE ? (y/n) :y
```

```
lager@server2:~>
```

Note: Backup files will never get deleted by *Stock Manager*.

4.7 Macro Functions

Up to now, we addressed individual entries of items. In certain cases this would be very time consuming and error prone. Situations which require efficiency are for example:

1. query data base for availability of parts by a facility BOM
2. withdrawing parts from stock by a facility BOM
3. generating a withdrawal list for items to be taken from stock
4. generating an order list to purchase items from distributors

The starting point for this operations is the so called facility BOM.

4.7.1 Generate Facility BOM

The facility bill of material is a csv formatted file that contains the components to be populated on the PCB. Instead of much words, just open the example provided here:

https://github.com/Blunk-electronic/stock_manager/blob/master/example_database/facility_bom.csv

There you find quantities, part IDs, prices and some statistic information for ONE unit or board to be populated. But how to you get this file ?

Answer:

1. writing by hand (not recommended)
2. generating with *Stock Manager*

Point 2 is what we prefer. Every CAE tool (like *CadSoft EAGLE*) can output a raw BOM file with basic information about part name, value and some properties.

Customize the CAE tool so that it outputs the columns and header as shown in this example:

https://github.com/Blunk-electronic/stock_manager/blob/master/example_database/bom_created_by_eagle_ulp.csv

This file is input into *Stock Manager* as shown below:

```
lager@server2:~> stock_manager make_bom bom_created_by_eagle_ulp.csv facility_bom.csv
```

```
STOCK MANAGER version 001
=====
action          : MAKE_BOM
facility name    : BEL
input eagle bom : bom_created_by_eagle_ulp.csv
output bom      : facility_bom.csv

BOM ready

lager@server2:~>
```

That's it. See the result here:

https://github.com/Blunk-electronic/stock_manager/blob/master/example_database/facility_bom.csv

The facility BOM generator checks if all parts are set up on stock, otherwise it aborts and notifies you about non-existing parts.

Now, you are ready for macro functions.

4.7.2 Querying Items by Facility BOM

Before launching the assembly, one has to make sure, all parts are available on stock. In case of a shortage, an order list will be generated. So if you intend to assemble 1 unit/board, see following example:

Example #1:

```
lager@server2:~> stock_manager query_bom facility_bom.csv 1
```

```
STOCK MANAGER version 001
```

```
=====
action          : QUERY_BOM
input fac. bom  : facility_bom.csv
order list      : facility_bom_order.csv
qty of units    : 1
```

All positions of given BOM are available on stock.

No order list created.

```
lager@server2:~>
```

Everything is fine. Now lets assume there are 4 board to populate.

Example #2:

```
lager@server2:~> stock_manager query_bom facility_bom.csv 4
```

```
STOCK MANAGER version 001
```

```
=====
action          : QUERY_BOM
input fac. bom  : facility_bom.csv
order list      : facility_bom_order.csv
qty of units    : 4
```

Parts need to be ordered. Please read file 'facility_bom_order.csv' !

```
lager@server2:~>
```

An order list has been generated. It can be used to place orders at the distributors offered therein:

https://github.com/Blunk-electronic/stock_manager/blob/master/example_database/order_list.csv

4.7.3 Checking Out Items by Facility BOM

In contrast to *query_bom* , which is a mere probing, *checkout_bom* withdraws many items from stock simultaneously. *Stock Manager* is to be provided with facility BOM and quantity of units to populate.

First, the case of a shortage of parts is shown in

Example #1:

```
lager@server2:~> stock_manager checkout_bom facility_bom.csv 4
```

```
STOCK MANAGER version 001
```

```
=====
action          : CHECKOUT_BOM
input fac. bom  : facility_bom.csv
order list      : facility_bom_order.csv
withdrawal list : facility_bom_withdrawal.csv
qty of units    : 4
```

```
Parts need to be ordered. Please read file 'facility_bom_order.csv' !
```

```
Withdrawal list not created.
```

```
No parts checked out from stock.
```

```
lager@server2:~>
```

As long as parts are not available, no checkout will be done. An order list will be generated. Hence assembly can not commence. You are forced to order parts, to update quantities (see section 4.2.1.1 Quantity on Stock page 12) and finally to try again.

Now we try to checkout parts for only ONE unit.

Example #2:

```
lager@server2:~> stock_manager checkout_bom facility_bom.csv 1
```

```
STOCK MANAGER version 013
```

```
=====
action          : CHECKOUT_BOM
input fac. bom   : facility_bom.csv
order list       : facility_bom_order.csv
withdrawal list  : facility_bom_withdrawal.csv
qty of units     : 1
```

All positions of given BOM are available on stock.

No order list created.

WARNING: All positions of given BOM will be checked out from stock !!!

WARNING: After checkout you must take all items from stock as listed in file
'facility_bom_withdrawal.csv' !

ARE YOU SURE ? (y/n) :y

Now withdraw items from stock as listed in file 'facility_bom_withdrawal.csv' !

```
lager@server2:~>
```

The result is obvious. No parts to order, but a withdrawal list has been generated. All quantities affected have been updated now. According to this list

https://github.com/Blunk-electronic/stock_manager/blob/master/example_database/withdrawal_list.csv

you are now required to take those items from stock as soon as possible.

4.8 Getting Help

Whenever you need help or example commands just type

```
stock_manager help
```

which outputs a list of example commands with short explanations.

You can download these help files from here:

https://github.com/Blunk-electronic/stock_manager/blob/master/conf/.stock_manager/help_english.txt

https://github.com/Blunk-electronic/stock_manager/blob/master/conf/.stock_manager/help_german.txt

Finally, if you get stuck, please contact us via email or phone. See contact data on cover sheet of this document.

5 Facility Part Code Syntax

The current implementation of *Stock Manager* is tailored for EMS (Electronic Manufacturing Services). It can be adopted to fit into other fields like medical, transportation, food supply chains, customer data, ...)

Electronic components handled within such a company require a certain nomenclature. In the past a nomenclature evolved that deserves to be described here in detail.

Lets take the example of part with ID 13 we spoke about earlier. Its facility part code reads:

C_PAC_S_0805_VAL_100n

For an electronics specialist there is not much to explain:

1. The facility part code always starts with the prefix C, R, L, IC, LED, DIS, X, J, S, K and so on. It indicates what category of component it belongs to (capacitor, resistor, ...)
2. The next field is a reserved keyword PAC_S or PAC_T which announces the next field is the package name. In this case it is a SMD 0805 package.
3. Keyword VAL indicates that the next field is the actual value of the capacitor (100 Nanofarad)

These are the minimum parameters that describe a capacitor. If more properties are to define another example might help.

Part 40 has a facility code that reads

C_PAC_T_RADIAL_7mm_GRID_2.54mm_VAL_100u_Vmax_35V

1. It is a capacitor.
2. Its package is a THT type with a radial housing of 7mm diameter
3. pin grid is 2.54mm (0.1 inch)
4. value 100 Microfarad (note the "u" instead of "μ")
5. maximum voltage rated is 35V

You are free to define more properties like maximum temperature, temperature coefficient, losses and so on. Important are the keywords that announce a property. Take your time to have a look at other facility part codes used in the example stock data base provided at

https://github.com/Blunk-electronic/stock_manager/blob/master/example_database/stock_db.csv

Inside the CAE tool the facility part code is to be assigned for individual parts of the design. The facility part code serves as primary key all the way from the design down to the stock data base, order list and withdrawal list.

6 Configuration Files

In order to do its work properly, *Stock Manager* requires a bit configuration according to the facility it is deployed.

A hidden directory `.stock_manager/` that lives in the home directory of the operator contains the help files (see section 4.8 Getting Help page 29) and most important a configuration file `stock_manager.conf`. Find an example here:

https://github.com/Blunk-electronic/stock_manager/blob/master/conf/.stock_manager/stock_manager.conf

This configuration file is easy to understand:

```
language english
currency EUR
# The first facility name will be used as primary.
# Others serve as alternatives in the order they appear here.
# If primary name not available in eagle bom file, alternatives will be checked.
facility_name BEL JEN

#directories
directory_of_stock_data_base /home/luno/controlling/stock
directory_of_log              /home/luno/controlling/stock/log
directory_of_backup           /home/luno/controlling/stock/bak

#customer specific prefixes NOTE: Affected parts require a function assigned to !
customer_prefixes            XS XP XD
```

Here you define the language of the help file, the currency and your facility name. Describe your facility with maximal three upper case letters (similar to the way airports are designated like FRA or TXL). For example

"Blunk electronic" can be abbreviated with BEL.

The last three lines define where the stock data base file lives, where the log data goes and where the backup files go. Note, here you define directories (or folders) which can be local, NFS or Samba shares. It is highly recommended to include those directories in your regular corporate backup scheme.

Assigning directories via environment variables like `$HOME/controlling/stock` is not supported currently.

7 Installation

For UNIX or Linux administrators, compiling and installation should not pose serious problems. There are no libraries required. Copy the binary file *stock_manager* in \$HOME/bin and modify the \$PATH variable.

However, feel free to contact us as shown on the cover page of this paper.

Read section 11 To Do page 35 for open issues.

Please find releases and source code at:

https://github.com/Blunk-electronic/stock_manager

7.1 Stand Alone

If only one user is supposed to work with *Stock Manager* no safety measures are required.

7.2 Server

If a group of people is allowed to work with *Stock Manager* it is **highly recommended** to:

1. Setup a user on the server machine who is allowed to write/read access to the stock data base.
2. Ensure only this very user is allowed to log in only once on the server.
3. Ensure only this very user is allowed to run the binary *Stock Manager*.

8 Limitations

8.1 Manufacturers

Currently there is support of up to 2 manufacturers of an item.

8.2 Distributors

Currently there is support of up to 6 manufacturers of an item.

8.3 Maximum Number of Item Entries in Stock Data Base

There is an upper limit of entries possible in the stock data base. This limit is set by the workstation RAM and swap space size.

8.4 Loading and Saving Time

Since *Stock Manager* is a single executable Linux file, it loads and saves the stock data base on every launch. In slow networks this might cause congestion and excessive delays.

9 Extendability

As pointed out in section 3 The Concept Behind on page 6, the program can work together with third party software. Extensions of this kind are possible:

1. batch execution or scripting
2. spawning from other software
3. graphical user interface (GUI)
4. interfacing with SQL data bases

10 Frequently Asked Questions

Readers used to MS-Excel might raise the question: "Why don't you use xls files ?".

Answers:

- ➔ xls is a non-disclosed proprietary format of Microsoft
- ➔ it is binary coded
- ➔ it can be generated by MS software only
- ➔ it can not be read properly 10 years later by newer versions of MS-Excel
- ➔ MS does not qualify for mission critical purposes

Why programming a stock management safety critical in Ada ?

- ➔ Populating PCBs with electronic components itself is safety critical.
- ➔ Wrong populated PCBs cause excessive loss of time, money and material.
- ➔ Electronic components and PCBs targeted for avionics, transportation and medical are safety relevant.

11 To Do

1. provide a summary and statistics of stock data base on commands
2. print currency after prices
3. translate help files in other languages
4. allow comments in configuration file
5. implement file lock mechanisms
6. ensure safe operation if first item is entered and stock data base is non-existent
7. create log/bak directory if non-existent
8. write proper Linux Man-Page
9. support time zones
10. improve "no_confirmation" function
11. implement show_by_storage_place function
12. ensure storage place can be assigned only once. Probably not a good idea. This would forbid using the same storage place for more than one item.
13. improve facility part code check
14. ensure manufacturer data can be edited, only if manufacturer name has been assigned
15. log rollback operations
16. install script

12 Links

- (1) *CadSoft EAGLE Consulting* – a reasonable way to reasonable work at <http://www.blunk-electronic.de>

- (2) The office alternative : *LibreOffice* at <http://www.libreoffice.org>



- (3) *EAGLE* - an affordable and very efficient schematics and layout tool at <http://www.cadsoftusa.com>



- (4) Looking for a lean **Boundary Scan Test System** ? Please have a look **here !**

13 Disclaimer

This document is believed to be accurate and reliable. I do not assume responsibility for any errors which may appear in this document. I reserve the right to change it at any time without notice, and do not make any commitment to update the information contained herein.