[Cormat\_py] Requirements Specification

Version 1.1

Last edited on January 29, 2019

Author: bruno.viola@ukaea.uk

Table of Contents

[1. Executive Summary 3](#_Toc535503769)

[1.1 Project Overview 3](#_Toc535503770)

[1.2 Purpose and Scope of this Specification 3](#_Toc535503771)

[2. Product/Service Description 3](#_Toc535503772)

[2.1 Product Context 3](#_Toc535503773)

[2.2 User Characteristics 3](#_Toc535503774)

[2.3 Assumptions 4](#_Toc535503775)

[2.4 Constraints 4](#_Toc535503776)

[2.5 Dependencies 4](#_Toc535503777)

[3. Requirements 4](#_Toc535503778)

[3.1 Usability 11](#_Toc535503779)

[3.2 Performance 11](#_Toc535503780)

[3.2.1 Capacity 11](#_Toc535503781)

[3.2.2 Availability 12](#_Toc535503782)

[3.2.3 Latency 12](#_Toc535503783)

[3.3 Manageability/Maintainability 12](#_Toc535503784)

[3.3.1 Maintenance 12](#_Toc535503785)

[3.3.2 Operations 12](#_Toc535503786)

[3.4 System Interface/Integration 12](#_Toc535503787)

[3.4.1 Protection 12](#_Toc535503788)

[3.5 Data Management 12](#_Toc535503789)

[3.6 Standards Compliance 12](#_Toc535503790)

[3.7 Portability 12](#_Toc535503791)

[4. User Scenarios/Use Cases 12](#_Toc535503792)

[5. Deleted or Deferred Requirements 12](#_Toc535503793)

[APPENDIX 13](#_Toc535503794)

[Appendix A. Definitions, Acronyms, and Abbreviations 13](#_Toc535503795)

[Appendix B. References 13](#_Toc535503796)

[Appendix C. Organizing the Requirements 13](#_Toc535503797)

# Executive Summary

## Project Overview

The program Cormat\_pyis written in Python 3.5, for the manual/automatic correction of fringe Jumps (FJ) in KG1V data.

## Purpose and Scope of this Specification

This code is intended to be used by KG1V ROs and other people authorized to handle KG1 data.

In scope

This document is intended as being an introduction to the Cormat\_py code, with reference to further documentation on KG1 system and the KG1R\_py code as required. It is also a snapshot of the code in its current state (as January 2019), that will help with any further development as required. The code can be found on Gitlab [here](https://git.ccfe.ac.uk/bviola/Cormat_py). Documentation has not been released yet.

Out of Scope

It is outside the scope of these specification to describe how KG1 system works

# Product/Service Description

The Cormat\_py code is a new code written in Python designed to eventually replace the current CORMAT code. CORMAT code is used to manually validate KG1 PPF data for the line-integrated density of the 8 different KG1 channels

This new code has to be released as soon as possible as to replace CORMAT.

## Product Context

CORMAT\_PY code is related to KG1\_py , CORMAT and KG1V\_IDL codes. The old KG1V\_IDL chain1 code reads in JPF data from the KG1V data acquisition system. However, the KG1V hardware is very old and requires regular restarts, impacting on plasma operations. Therefore, two new data acquisition systems have been commissioned to replace KG1V : KG1RT and KG1C. KG1R\_PY has been developed to process the JPF data from these systems, to produce PPF data for the line-integrated densities and correct fringe jumps.

CORMAT\_py will be able to process data coming from KG1V and the newer KG1RT and KG1C systems.

## User Characteristics

Create general customer profiles for each type of user who will be using the product. Profiles should include:

The users of this code are:

* KG1 Responsible Officer (RO): is the responsible of the diagnostic. This user is authorized to write public KG1 ppf. He authorizes changes to hardware and software related to KG1.
* Users authorized to write KG1 public ppfs: users that after training have been authorized to write KG1 public data and can correct frinje jumps in the signals. Some of these users can make changes to software related to KG1
* Trainees: users that are training to manipulate KG1 data. They are not allowed to write public KG1 ppfs, only private PPFs.

## Assumptions

Users of these code are required to have expertise with fringe jump correction. In particular they:

* Know what a FJ is
* Know what signal can help identify one
* Know how to manually correct it

The code is written in Python 3 and will run on the Freja cluster using python/3.5.1.

## Constraints

* The code:
* Is written in Python 3 (ver 3.5.1)
* will operate with data coming all KG1 acquisition systems
* will be run only by users authorized by KG1 RO

## Dependencies

* This code will require a secure connection to Freja cluster to run

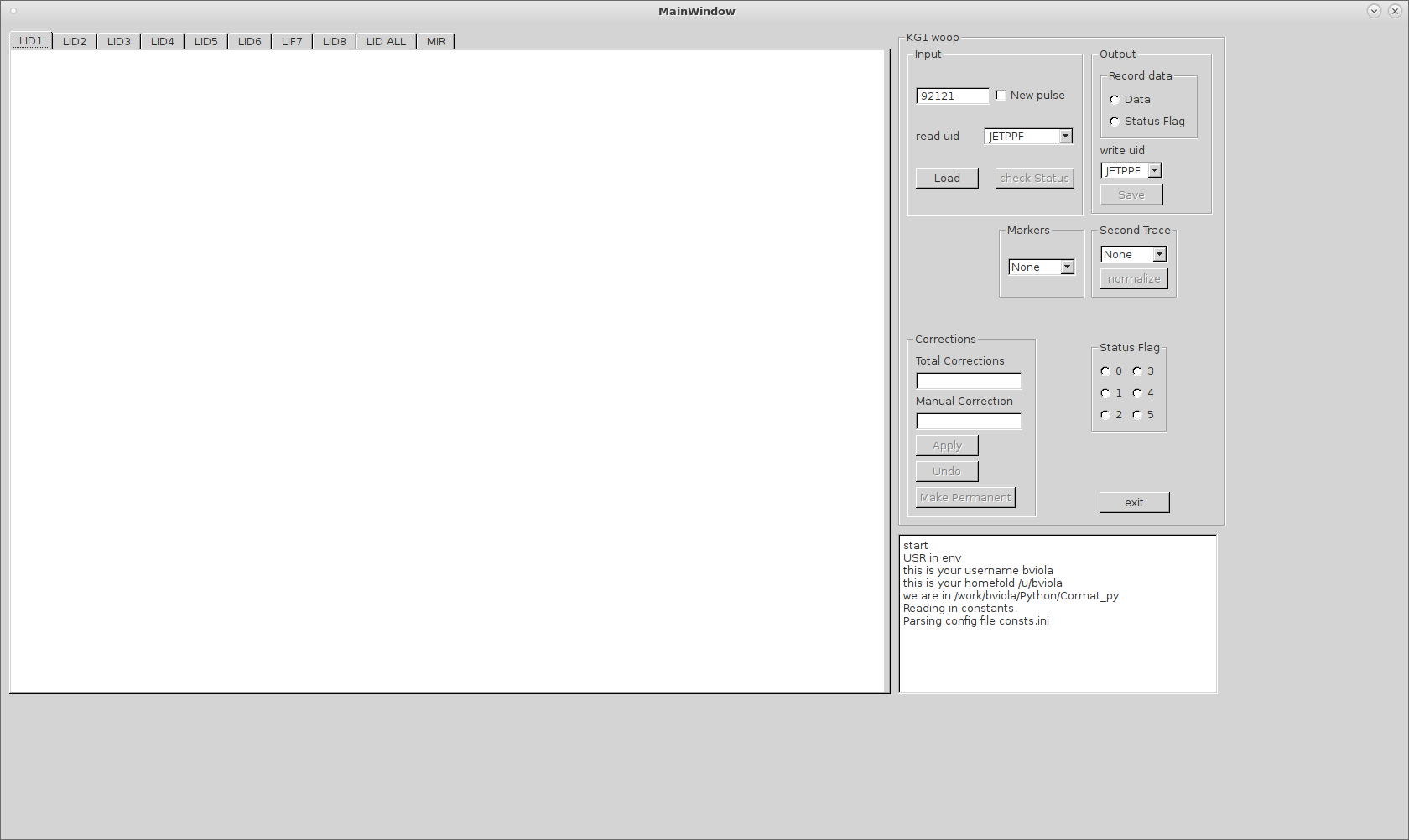
The current design of the graphical interface (GUI) is the following:

Figure 1:Cormat\_py Main Window

Each tab has a Navigation bar as Fig.2

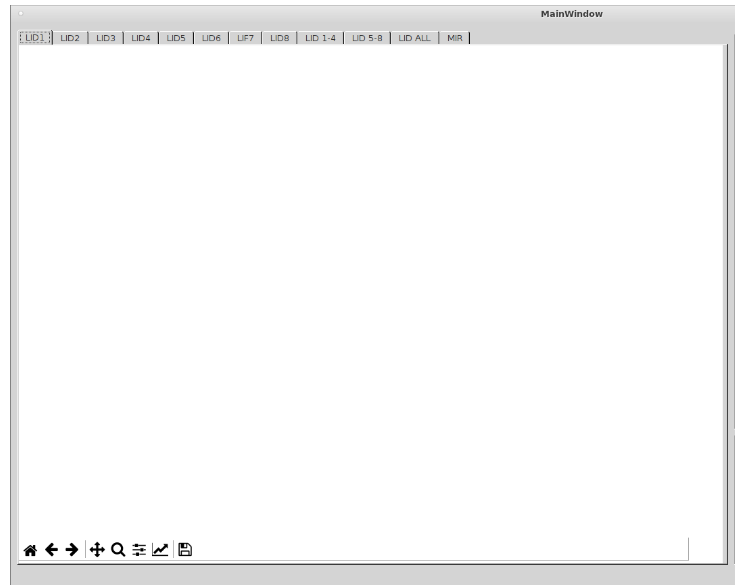


Figure 2: Tabs and Navigation bar

The Main menu is as in Fig.3

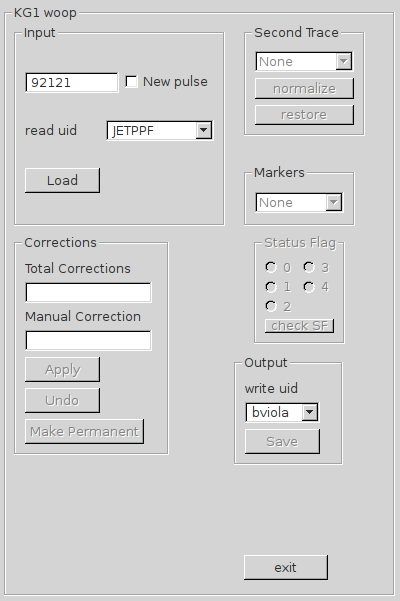
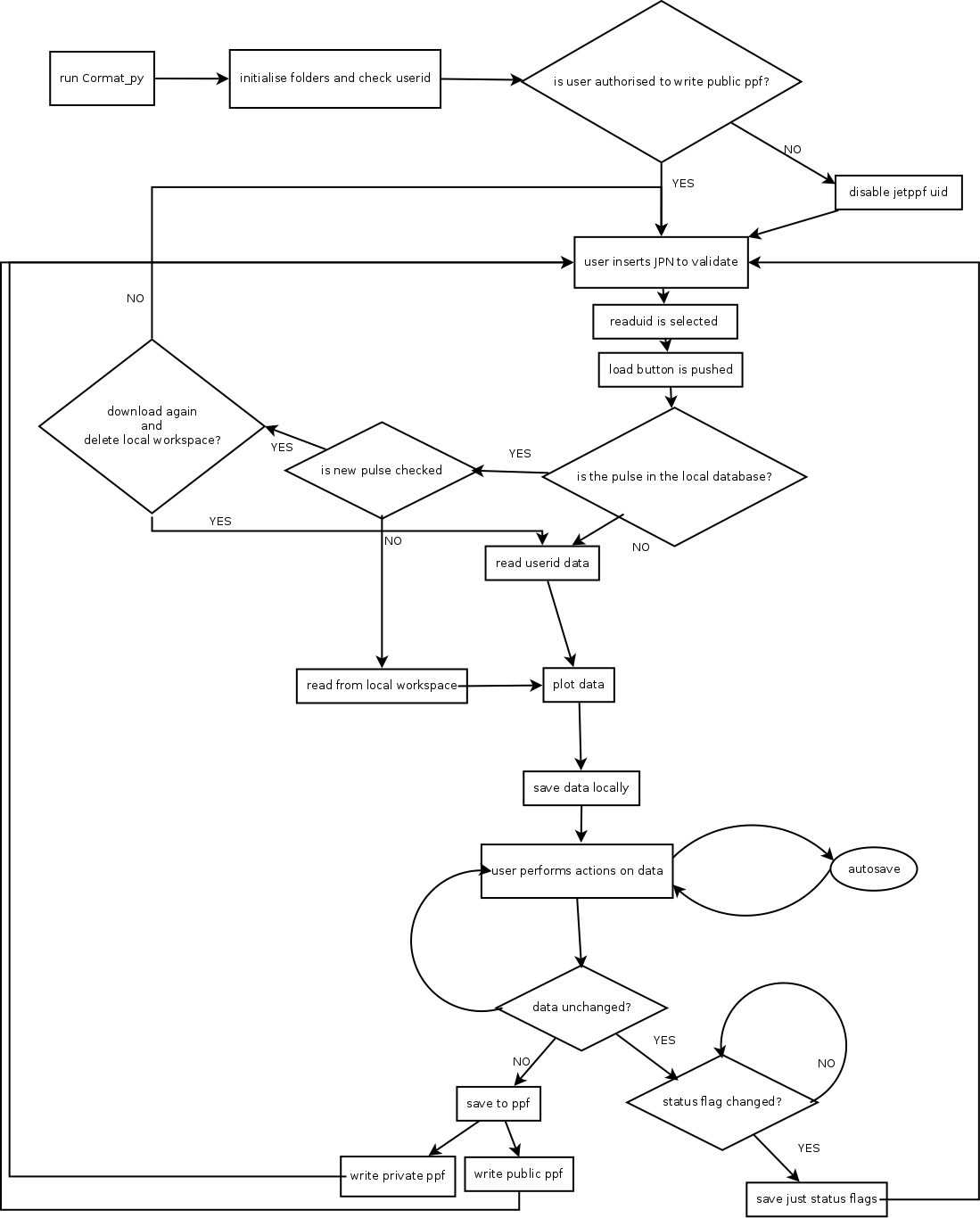


Figure 3: main menu

# Flow Chart



# Requirements

Priority Definitions

The following definitions are intended as a guideline to prioritize requirements.

* Priority 1 – The requirement is a “must have”
* Priority 2 – The requirement is needed for improved processing, and the fulfillment of the requirement will create immediate benefits
* Priority 3 – The requirement is a “nice to have” which may include new functionality

| Req# | Requirement | Comments | Priority | Date Rvwd | Reviewed / Approved |
| --- | --- | --- | --- | --- | --- |
| 1 | Will operate with data coming from all KG1 acquisition systems (kg1v,kg1r and kg1c) | Tested with 81472 and 92121 | 1 | 15/01/19 |  |
| 2 | user interface that has 12 tabs: one for each of the 8 channels, one for lateral channels only, one for vertical channels only, one for all the channels and one for the mirror movements |  | 1 | 15/01/19 |  |
| 3 | user is required to input the pulse number that wants to process in the green lineEdit |  | 1 | 15/01/19 |  |
| 4 | GUI needs a checkbox “new pulse” that, when clicked, makes the code download again the data |  | 1 | 15/01/19 |  |
| 5 | There must be an internal check to avoid deletion of previous loaded and unsaved data, i.e. data has changed during FJ correction | . | 1 | 15/01/19 |  |
| **6** | Load button reads the following data: KG1, KG4, NBI data, Elm data,Lidar data, Pellet, HRTS, JxB, ICRH, LHCD, MIR, Prad |  | 2 | 15/01/19 |  |
| **7** | Data is saved if reading is successful in binary files (pickle format):  *Pulsename\_kg1.pkl* that contains just KG1 raw data  *Pulsename\_kg1.pkl* that contains just KG1 data (for changes)  *Pulsename.pkl* that contains the remaining data | Add autosave 1-5min | 1 | 15/01/19 |  |
| 8 | Code must have a check on *Pulsename\_kg1.pkl* file attributes as eventually this file is going to be modified and the user doesn-t want to lose this information. |  | 1 | 15/01/19 |  |
| 9 | The check on Pulsename\_kg1.pkl must happen also when reading pulse data to avoid overwritin sensible information |  | 1 | 15/01/19 |  |
| 10 | Code must keep a log of all the processed data and tracing/cancel changes |  | 1 | 15/01/19 |  |
| 11 | Code must save locally all processed data |  | 1 | 15/01/19 |  |
| 12 | user can select to read data from a list of “authorized users”: this is a list of user chosen by the RO that can write KG1 ppfs (red comboBox) |  | 1 | 15/01/19 |  |
| 13 | when user is done with data validation he can:  save data to ppf  save just Status flag |  | 1 | 15/01/19 |  |
| 14 | if the user running the code is a trainee (or is not in the list of authorized users) he will no be able to save public data. | Trainee can save only private data | 1 | 15/01/19 |  |
| 15 | Once data is loaded it will be displayed on the different tabs. Each channel goes on a tab, with the exception of multi-plots (see tabs for lateral channels only, for vertical channels only,for all the channels and for the mirror movements) |  | 1 | 15/01/19 |  |
| 16 | The user can select to super impose an additional trace choosing among: HRTS, LIDAR, BREMSTRAMLUNG, FAR, KG4 RT, Cutton-Mutton. KG1 RT (orange comboBox) |  | 1 | 15/01/19 |  |
| 17 | The user can select to super impose plasma markers on the tabs showing KG1 data. The marker are: elms, nbi time, current flat top, magnetic field greater than given threshold, nbi time, pellet time (yellow comboBox) |  | 1 | 15/01/19 |  |
| 18 | a zoom/reset function must be available for each tab |  | 1 | 15/01/19 |  |
| 19 | a normalize plot function must be available for each tab |  | 1 | 15/01/19 |  |
| 20 | user can be allowed to zoom and move the signal (mouse event or keyboard short cut) |  | 1 | 15/01/19 |  |
| 21 | once the point for FJ correction is identified the user receive suggestion on how to correct |  | 2 | 15/01/19 |  |
| 22 | the user can apply correction either with mouse click or keyboard shortcut |  | 1 | 15/01/19 |  |
| 23 | correction is applied using *apply* button |  | 1 | 15/01/19 |  |
| 24 | once correction is applied the signal must be re-plotted |  | 1 | 15/01/19 |  |
| 25 | correction can be undone or made permanent with clicks of buttons. |  | 1 | 15/01/19 |  |
| 26 | Multiple correction must be made available (either with mouse event or keyboard shortcut) |  | 1 | 15/01/19 |  |
| 27 | the user can input the correction he/she wishes to apply in a lineEdit box |  | 1 | 15/01/19 |  |
| 28 | the sum of the corrections must be 0 |  | 1 | 15/01/19 |  |
| 29 | the sum of the correction is displayed in a lineEdit box |  | 2 | 15/01/19 |  |
| 30 | user can set status flag for the selected channel using a radio button |  | 1 | 15/01/19 |  |
| 31 | the user can select with mouse the signal trace and set time-dependent status flag |  | 3 | 15/01/19 |  |
| 32 | once the user has finished for all channels he can save to ppf, choosing between private and public using a comboBox |  | 1 | 15/01/19 |  |
| 33 | all logging messages are displayed in the gui |  | 3 | 15/01/19 |  |
| 34 | Plot JxB in MIR subplot |  | 1 | 29/01/19 |  |

## Usability

## Performance

### Capacity

Include measurable capacity requirements (e.g., the number of simultaneous users to be supported, the maximum simultaneous user load, per-user memory requirements, expected application throughput)

The code can be run simultaneously by different users as each wil have its own temporary data.

### Availability

### Latency

## Manageability/Maintainability

### Maintenance

The maintenance of the code at the date of writing this document is of Bruno Viola.

The code has been written in modules and the graphical user interface has been created in QT designer.

The code is available on a GIT repository so, all users with access can pull from the repository and download their local copy of the code.

Only developers can push to the master branch

### Operations

## System Interface/Integration

### Protection

KG1 public data can be written only by authorized users.

Only the RO can authorized users.

The master branch of the repository is protected from accidental pulls.

## Data Management

## Standards Compliance

## Portability

# User Scenarios/Use Cases

# Deleted or Deferred Requirements

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Req# | Requirement | Status | Comments | Pri | Date Rvwd | Reviewed /Approved |
|  |  |  |  |  |  |  |

# APPENDIX

The appendixes are not always considered part of the actual Requirements Specification and are not always necessary. They may include

* Sample input/output formats, descriptions of cost analysis studies, or results of user surveys;
* Supporting or background information that can help the readers of the Requirements Specification;
* A description of the problems to be solved by the system;
* Special packaging instructions for the code and the media to meet security, export, initial loading, or other requirements.

When appendixes are included, the Requirements Specification should explicitly state whether or not the appendixes are to be considered part of the requirements.

1. Definitions, Acronyms, and Abbreviations
2. References
3. Organizing the Requirements