esoc

European Space Operations Centre

Robert-Bosch-Strasse 5

D-64293 Darmstadt

Germany

T +49 (0)6151 900

F +49 (0)6151 90495

www.esa.int

|  |
| --- |
| Manual / User Guide / Handbook |

TPF generation application from APF file

Approval

|  |  |
| --- | --- |
| Title TPF generation application from APF file | |
| Issue Number 1 | Revision Number 1 |
| Author Francesca Covella | Date 12/02/2021 |
| Approved By | Date of Approval |
|  |  |

Change Log

|  |  |  |  |
| --- | --- | --- | --- |
| Reason for change | Issue Nr. | Revision Number | Date |
|  |  |  |  |

Change Record

|  |  |  |  |
| --- | --- | --- | --- |
| Issue Number  1 | Revision Number  1 | | |
| Reason for change | Date | Pages | Paragraph(s) |
|  |  |  |  |

Distribution

|  |
| --- |
| Name/Organisational Unit |
|  |

Table of contents:

[1 Introduction 4](#_Toc65226053)

[1.1 Abstract 4](#_Toc65226054)

[1.2 Intended readership 4](#_Toc65226055)

[1.3 Pre-requisites for using *GENERATE TPF* 4](#_Toc65226056)

[1.4 Applicability 4](#_Toc65226057)

[2 Application 4](#_Toc65226058)

[2.1 Premise 4](#_Toc65226059)

[2.2 Objective 4](#_Toc65226060)

[2.3 Source Code 5](#_Toc65226061)

[2.4 Launching the Application 5](#_Toc65226062)

[2.5 Using the Application 6](#_Toc65226063)

[2.6 Configurable Options 13](#_Toc65226064)

[3 Automatic mode option 14](#_Toc65226065)

[4 Related documents 15](#_Toc65226066)

# Introduction

## ­­­Abstract

This document describes the use of the *GENERATE TPF* application and functions contained therein.

## Intended readership

This document is intended for IMCS operators, in particular the Spacecraft Operators (SPACON) for the Integral mission.

## Pre-requisites for using *GENERATE TPF*

As this document is intended for IMCS operators and in particular to the SPACONs, some previous knowledge of the SCOS-2000 IMCS is assumed. Namely, it is assumed that the reader of this user manual is already familiar with the structure of the APF, EPOS and TPF files, generated by the Flight Control Team and the Flight Dynamic Team during the Mission Planning. It is also assumed that the reader has a sound understanding of the IMCS used in the DCR of the Integral Mission, e.g. the Manual Stacks, the Command History displays and the other TM and TC Applications.

## Applicability

This document is applicable to the Integral MCS. But would also function effectively for XMM-Newton.

# Application

## Premise

The source code for the application runs on the server. The program can be run in two different modes: an interactive mode, which relies on inputs from the user and can be easily accessed from the Desktop (see section 3) and an automatic mode, which run in the background as a back-up measure (see section 5). The focus on this User Manual is on the interactive mode, as this mode relies on inputs from the user and requires a brief and concise explanation about how the Application behaves and what kind arguments are expects from the graphical user interface, even if the user will find that it is very much self-contained and self-explanatory.

## Objective

The objective of the Application is to offer a simple and intuitive tool to generate a particular or a series of TPF(s) to be loaded on the manual stack. These TPFs can be used to quickly and safely recover the planned routine operations in case of commanding interruption (for instance in case of communications errors). The Automatic mode is intended to be called by the IMCS Mission Planning System to automatically generate all relevant TPFs for an entire revolution and version number at the time the revolution is planned.

In case the TPFs were not already generated by the MPS, the tool can be used in interactive mode, in case a certain sequence of commands is not uplinked successfully due to communication issues or to similar reasons. In such a case it might be useful to easily generate and retrieve the TPF corresponding to the missed occurrence of the sequence of interest, containing the correct values for the commands’ parameters. Lastly, the application can be used to retrieve any past or future sequence, for other purposes, as long as the associated APF and EPO files are available.

## Source Code

This section is included for completeness and can be skipped by the SPACON, it is more relevant to the engineer responsible for maintaining the code.

The application consists of the following pieces of code:

1. **TPF\_ library.py** 🡪 a general and flexible class intended to customise and write a TPF file. This code is written following closely the structure of [5] in order to help the user to set all the mandatory (and optional) fields of the headers and of the parameters.
2. **INTXMM\_library.py** 🡪 a general library containing some functions related to the Integral and XMM missions, used for instance to perform some checks and general tasks, such as getting the latest time of one revolution or parsing the APF file associated to a specific revolution.
3. **APF2TPF\_3options.py** 🡪 this source code contains the specific information about the function of transforming an APF into a TPF. It retrieves the values from the APF dictionary, following the user inputs and it calls the function to write the file into the outputs folder.
4. **Integral\_logo.png**
5. **APF2TPF\_GUI.**py 🡪 this code contains all the graphical functionalities of the graphical interface. It is completely disconnected from the functionalities of the TPF and of the APF documents’ specifications, but it creates all the widgets and it links to them their functionalities as well as defining methods to change the values of some widgets attributes based on the user behaviour.
6. **APF2TPF\_client.py** 🡪 the main code which calls the other parts of the application. This is the code which has to be called to run the application. The flag after the code name (-a for automatic and -i for interactive) determines the mode.

## Launching the Application

To launch the application one should follow two simple steps. The first is to right click on the Desktop and select the option *SPACON SCRIPTS.* Then, in the drop down menu which appears (see Figure 1), one should click the option is *GENERATE TPF*.



Figure 1 – Workspace Menu Display

## Using the Application

This triggers the graphical user interface in Figure 2 as a result. On the top-left the instruction panel guides the user through the various steps. Hence, this user manual is written for completeness and can be easily understood before using the application, to know what to expect. The “select folder” buttons allows the user to pick the folder of the revolution and version of interest (Figure 3). After making sure the path has been updated correctly (Figure 4) one can press OK. Now the “load apf” button is enabled and can be pressed (Figure 5). The progresses are shown by the pink progress-bar. This buttons parses the APF and stores efficiently its content as a series of TPFs.

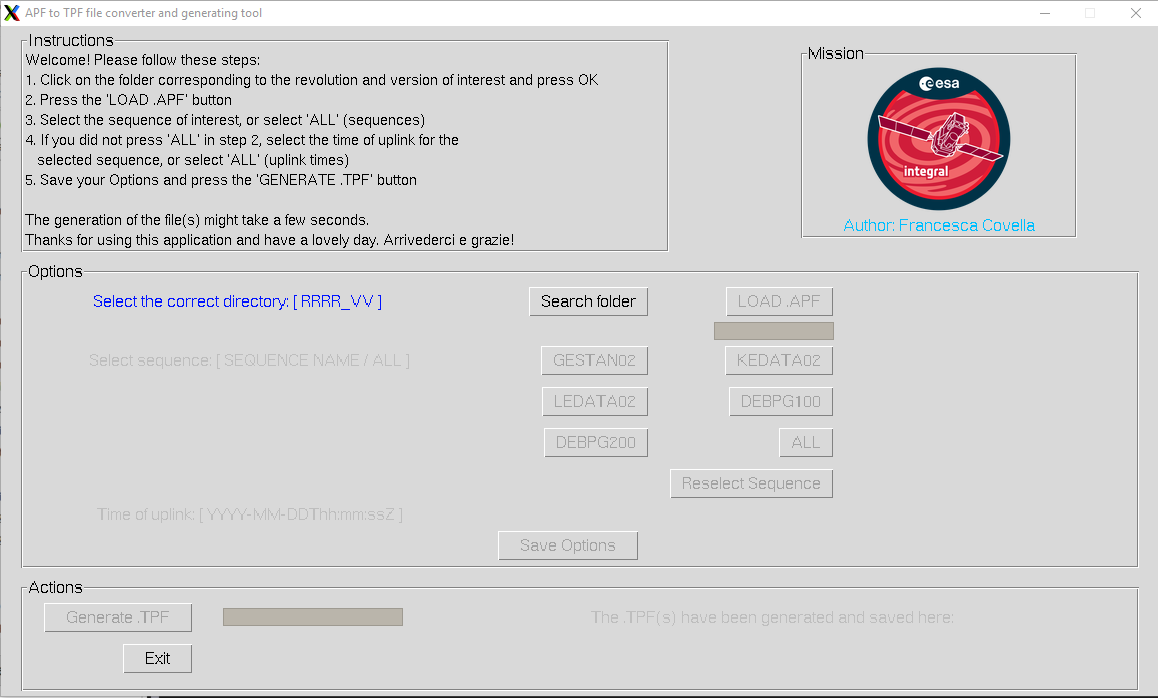


Figure 2: GUI pop-up

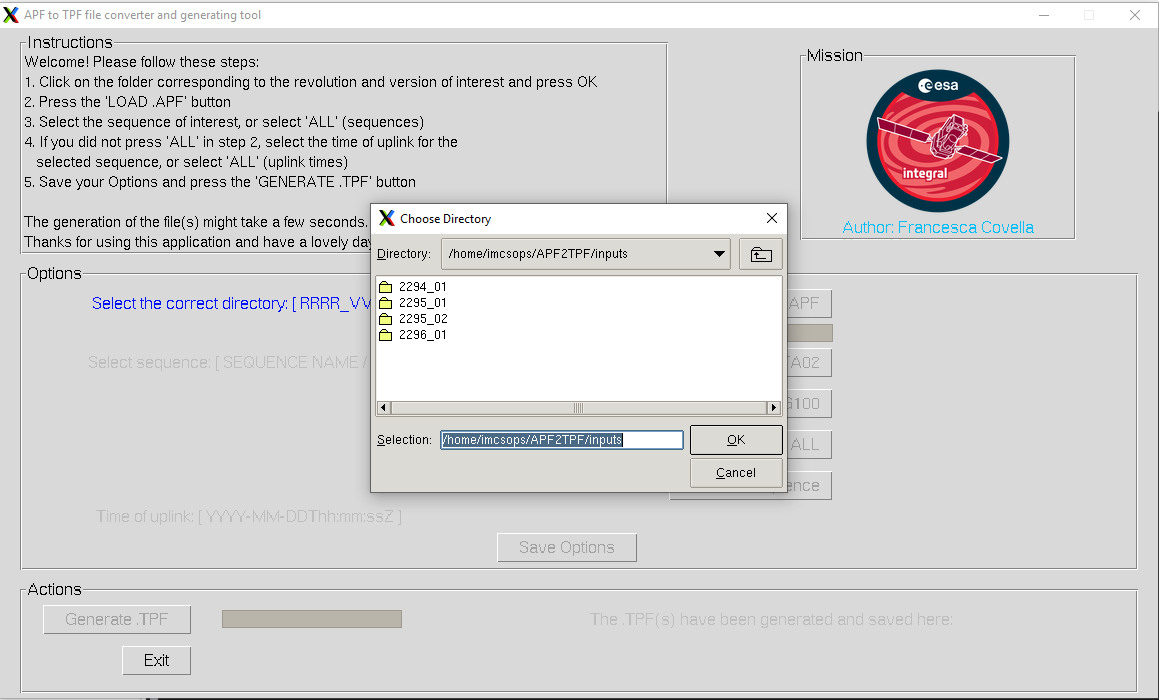


Figure 3: choose a directory

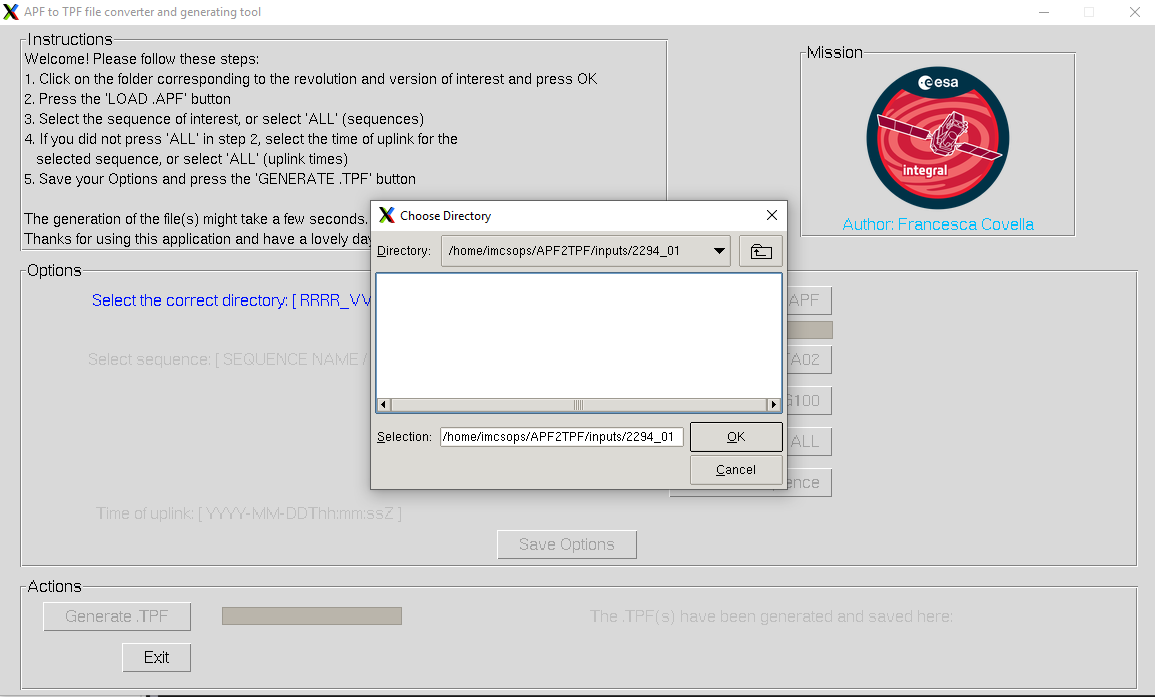


Figure 4: check you selected correctly the directory before clicking OK

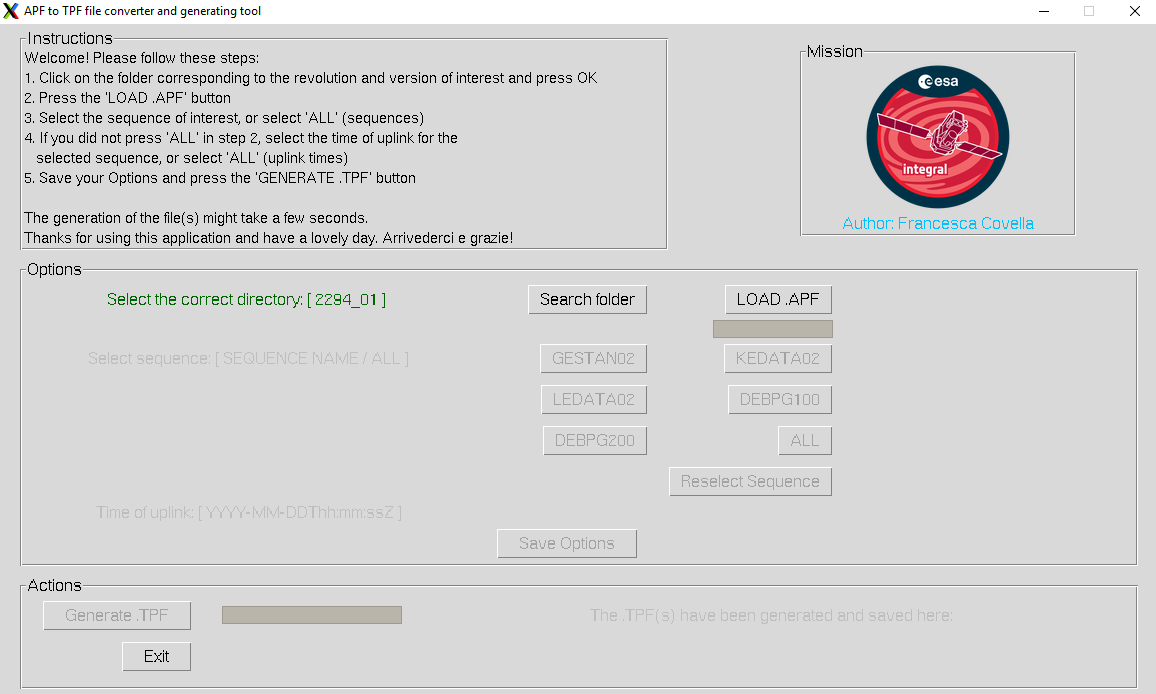


Figure 5: Load APF is enabled

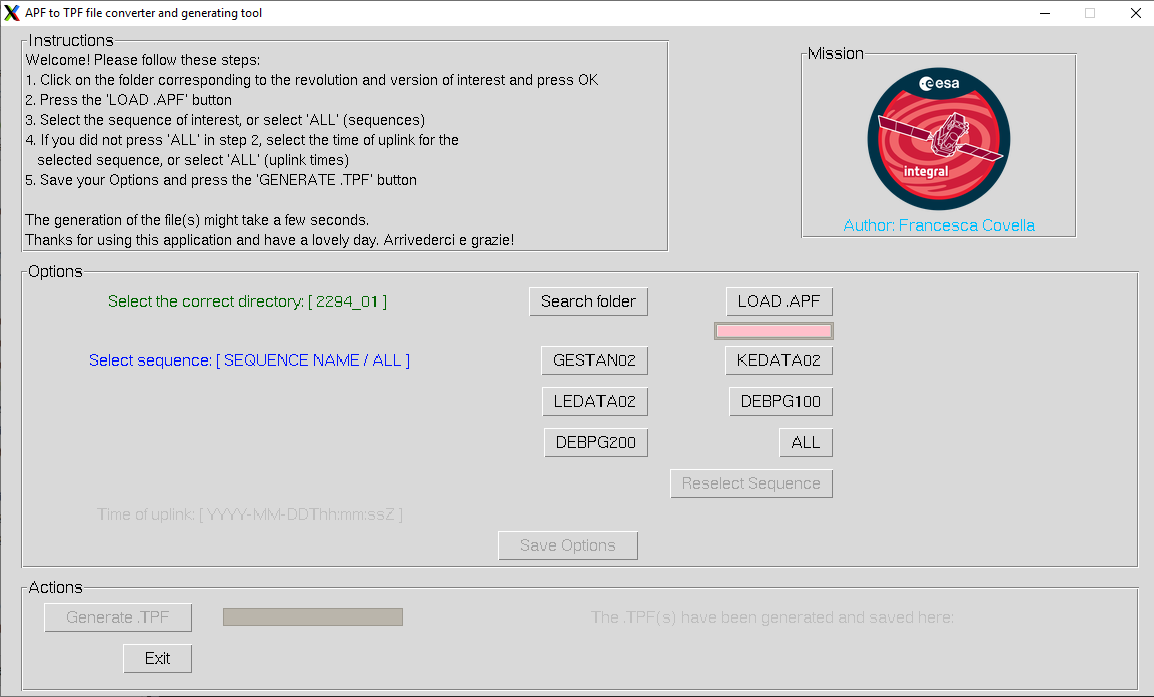


Figure 6: Select a sequence

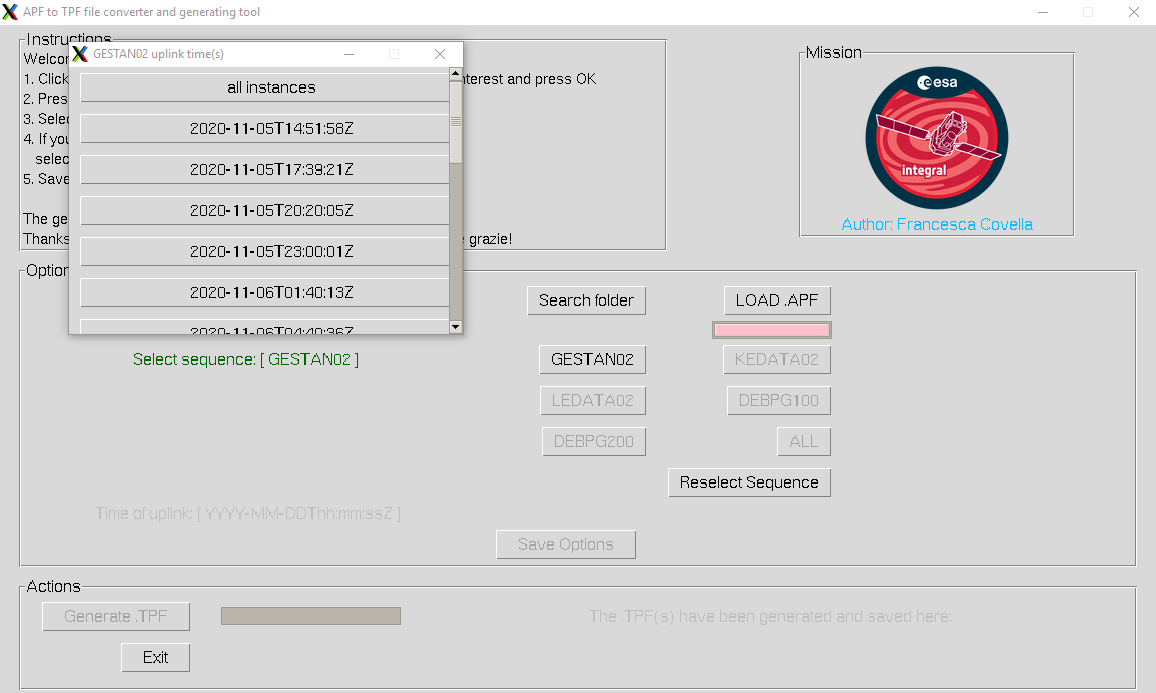


Figure 7: Time options menu pop-up

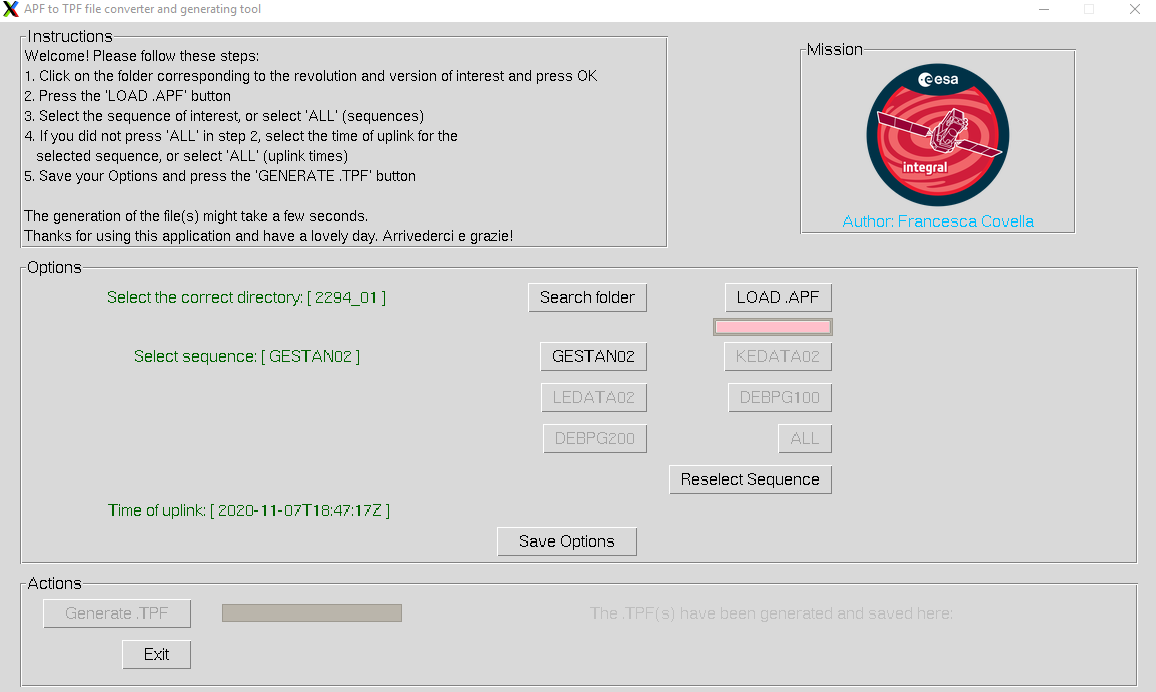


Figure 8: Choose to reselct the sequence (and time) or save settings

There are five possible sequences, for now, to choose from(Figure 6).

Upon selection of a sequence, a time pop-up menu appears (Figure 7).

One can select a single or all instances of the selected sequence based on planned release time and then close the menu. The selected option is diaplayed in green. There is a possibily to reselect the sequence and the time, if the user finds to have make a mistake. Note that if one selects “all” sequences, then the time field is automatically filled will “all”, as the TPF(s) of all instances of the selected sequences will be generated (Figure 9).

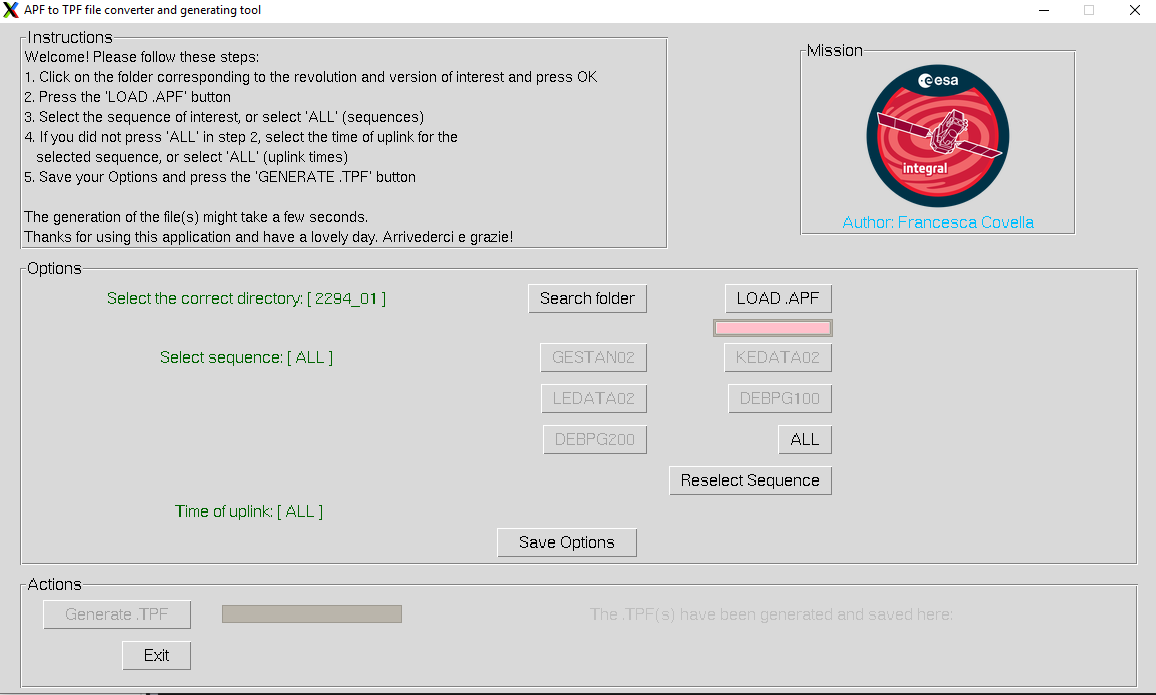


Figure 9: display after choosing ALL sequence

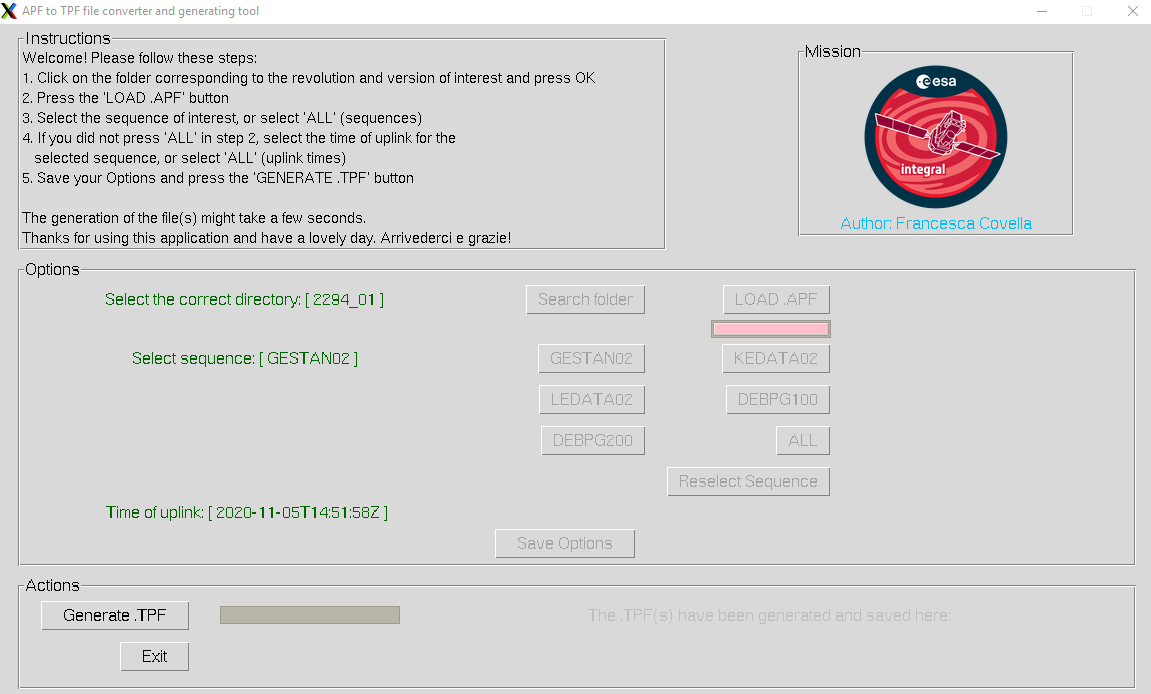


Figure 10: Ready to press Generate .TPF

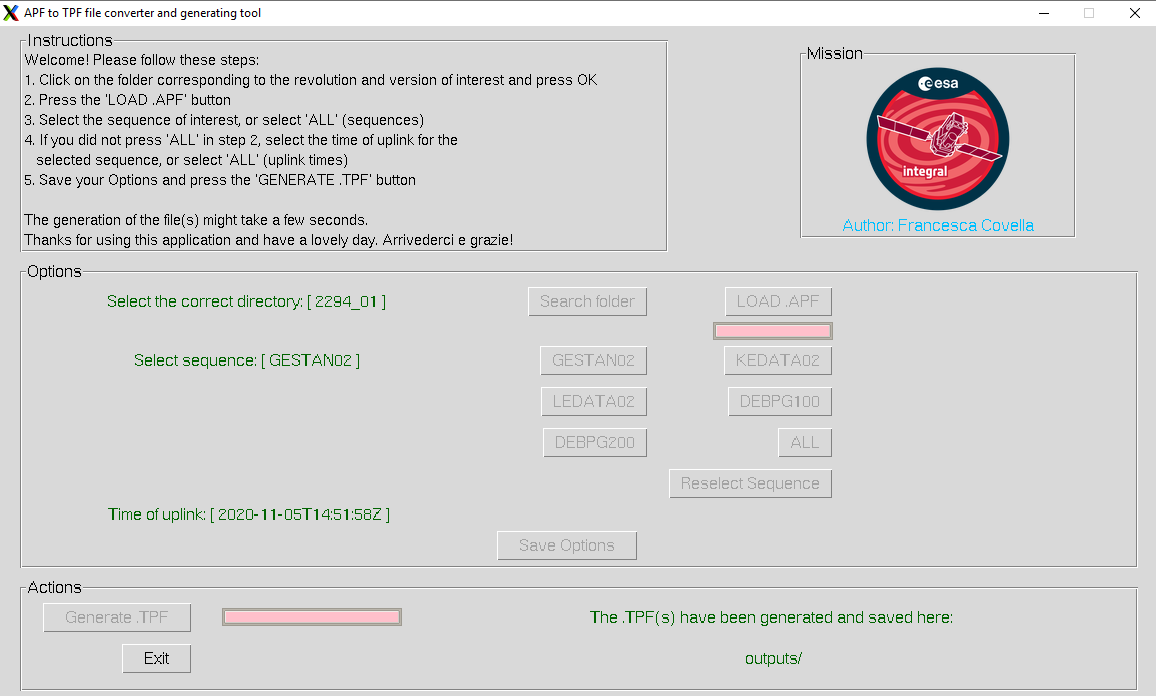


Figure 11: Ends of program

Once the user is sure of the selection, the button “save options” can be selected, which disables all the other buttons and enables the user to generate the TPF(s) (Figure 10).

The operation can be tracked by the progress-bar. Generating the file can take up to one minute if one choses to retrieve all the sequences at all times, otherwhise it would take a few seconds. Once the file(s) have been correctly generated, one can find the location printed out in green and can exit the program. The default location where the file will be saved is the prime server and the user will need to be download the TPF(s) in the outputs folder to a client.

## Configurable Options

This section is intended for the engineer responsible for maintaining the code and can be skipped by the user altogether.

There are a couple of paths that can be configured in order to control where to direct the user to pick from the possible directory containing the APF and EPO, upon selecting the “select directory” button, and where to save the generated file(s).

|  |  |
| --- | --- |
| Variable name  & location in the source code | Default option |
| input  APF2TPF\_client.py  Line 16 | The default path for the revolution\_version folders is scosii/MPS/data/  On the server |
| output  APF2TPF\_client.py  Line 17 |  |
| self.input  APF2TPF\_GUI.py  Line 86 | The default path for the revolution\_version folders is scosii/MPS/data/  On the server |
| self.output  APF2TPF\_GUI.py  Line 374 |  |

Table 1 – Configurable path options

# Automatic mode option

In the MPS client display, as shown below, the user can find the option of automatically retrieving **all** instances of all the sequence of interest from the .APF and generate a .TPF for each of them in the specified *todir* folder.

One of the TC application in the server or client?

The programme runs when a Timeline (TL) is authorized.

The automatic option is useful as a backup option and runs automatically after each timeline is ready, prior to the beginning of a certain revolution. It retrieves all the instances of all the sequence for a certain revolution and version number. These are specified in the first and second column of the table shown in figure 19.

RS: could even be default? We should configure a file clean though

How to keep track of the files as are produced (the code should display in the terminal the files as they are created?) I would propose to write them to a TPF sub-directory ..CMD/TPF/RRRR\_VV\_Generate\_TPF\_auto or similar

Where to find what it produced, always in the output folder, but how does the user know it?

Where to find the log if there has been any error Write the log to the same directory?

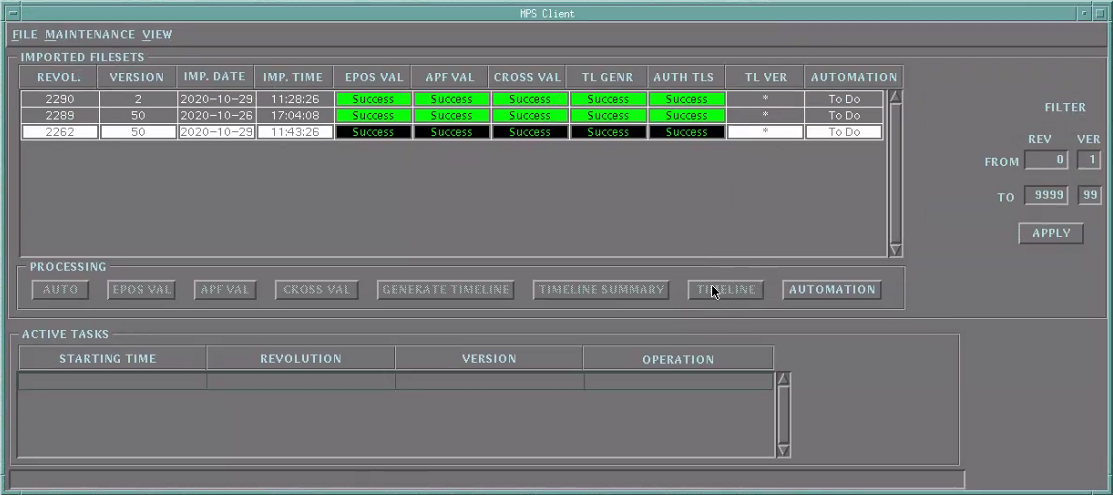


Figure 12 – File maintenance view Display

# Related documents

|  |  |  |
| --- | --- | --- |
| **#number** | **Document Reference Number** | **Title** |
| 1 | S2K-MCS-GLO-0001-TOS-GIC SCOS-2000 Glossary | Definitions and Acronyms |
| 2 | S2K-MCS-CF-0001-TOS-GIC SCOS-2000 | Configuration File |
| 3 | ESA PSS-04-107 | Packet telecommand standard, Issue 2 |
| 4 | ESA PSS-07-101 | Packet Utilisation Standard, Issue 1 |
| 5 |  | APF ICD |
| 6 | S2K-MCS-ICD-0003-TOS-GCI | SCOS-2000 Task Parameter Interface  Control Document |
| 7 | S2K-MCS-GLO-0001-TOS-GCI | Glossary, Definitions and Acronyms |
| 8 |  | EPO ICD |
| 9 | S2K-MCS-SUM-2310-TOS | SCOS-2000 Manual Stack Operator User Manual |
| 10 | S2K-MCS-SUM-2360-TOS-GIC | SCOS-2000 Command History Operator User Manual |

Table 2 – Important documents for understanding this User Manual