1. Overview

This document describes the usage and installation of the DPF Manager.

In the first section it is explained how to install the software in different operating systems.

The second section provides detailed explanation of the modes from which the application can be executed, i.e. **command line**, **graphical user interface** and **client/server**.

2. Installation Procedure

In this section the installation procedure for Windows, Linux and Mac operating systems are explained in detail.

In each case, the installer comes in two versions, one that includes the Java Virtual Machine (larger size) and another that does not (“lite” version). If the installer that does not include the JVM is chosen, then Java 8 needs to be already installed in the computer.

2.1 Java 8

Java 8 only needs to be installed for the versions of DPF Manager that do not include the Java Virtual Machine.

To check if Java 8 is already installed, open the terminal (in Windows OS `c:\Windows\System32\cmd.exe`) and type “java -version”. If the command is not found or the Java version is lower than 1.8, then Java needs to be upgraded.

Java 8 can be downloaded from the following URL: [http://www.oracle.com/technetwork/java/javase/downloads/index.html](http://www.oracle.com/technetwork/java/javase/downloads/index.html)

Go to the JRE link (or JDK if you are a developer), download the file for your operating system, run the installer and follow the steps in the screen.

You can also use the OpenJDK (instead of the OracleJDK). However, if you want to use OpenJDK, you will need to build with the open version of OpenJDK that includes JavaFX that can be downloaded from here: [https://wiki.openjdk.java.net/display/OpenJFX/Building+OpenJFX](https://wiki.openjdk.java.net/display/OpenJFX/Building+OpenJFX)

In Linux you may need to set the environment variable JRE_HOME and point it to the location where Java has been installed.
2.2 Windows

2.2.1 Installation

The Windows installer is provided in a .exe file. A version that includes the Java Virtual Machine is available for 64-bit machines. And a version that does not include the Java Virtual Machine is available for 32 and 64 bit machines.

The EXE installer is a four-step installation process. Open the installer, and click the “Next” button. Then select the desired folder for the application to be installed and click “Next”.

After that, specify the application name and click “Next”. Finally, in the last step you can select whether to create a desktop shortcut or not, and also if you want to add the DPF Manager in the system path variable (so that you can run the program with the terminal without specifying its full path) and then click “Next” and “Install” for installing the software in the computer. At the end of the installation wizard you can choose to start the installed program.

A link to the application will be put in the Windows Start Menu and also in the desktop (if the option in the installer was selected). The default location of the user data directory (where the reports and configuration files are stored by default) is: "c:\users\YOUR_USER\DPF Manager".
2.3 Linux

For Linux installation, there are installers in .deb and .rpm formats, so as to be used in the most common Linux distributions, including Ubuntu, Fedora, Debian and Open Suse.

2.3.1 DEB installer

To install the .deb package double click it and follow these steps:

Click the “Install” button, in the top right.
When prompted, click “Ignore and Install”. After that, enter your root password to grant administrator privileges. Now the DPF Manager is installed in the computer. Its default location is “/opt/DPFManager”.

In order to run the DPF Manager in Linux, remember to have set the JRE_HOME variable and pointing to the actual Java location.

### 2.3.2 RPM installer

Double click the install button when asked and wait until the packages have been installed.

The DPF manager folder is located in the default location “/opt/DPFManager”.

In order to run the DPF Manager in Linux, remember to have set the JRE_HOME variable and pointing to the actual Java location.

### 2.4 MacOS

For Mac OS, the installer is a DMG file. For installing the DPF Manager in Mac OS, double click the DMG, and then just drag the application and drop it into your Applications folder, as indicated.

To use the DPF Manager in Mac with the terminal, the executable is located at “/Applications/DPF Manager.app/Contents/MacOS/DPF Manager”.

You can create an alias to run it from anywhere with the command:

```
alias dpf_manager="/Applications/DPF Manager.app/Contents/MacOS/DPF Manager"
```
3. Application Usage

DPF Manager can be run in command line mode (CLI) and through a Graphical user interface (GUI).

In Linux there is a single executable for both interfaces of the DPF Manager, with the name “dpf-manager”. In order to open the GUI, execute “dpf-manager” (without parameters). For running the CLI, execute it with parameters, e.g. “dpf-manager --help”.

In Windows there are two executables, one for the CLI, named “dpf-manager-console.exe” and another one for the GUI, named “DPF Manager.exe”.

For MacOS, there is a single package in the Applications folder with the name “DPF Manager” that runs the GUI by default (double-clicking it). The CLI can be run by executing the inner executable “DPF Manager.app/Contents/MacOS/DPF Manager” through the terminal.

3.1 Command line

To run the DPF Manager from the command line, execute the appropriate executable file for your operating system as explained before (in this section we will refer to it simply as dpf-manager, as it is in Linux), and run the following command:

dpf-manager [commands]

These are the available commands:
- check: Performs a local files check.
- config: Manages the configuration files.
- gui: Launches graphical user interface.
- modules: Manage the conformance checkers.
- periodic: Manage periodical checks.
- remote: Performs remote file checks (client mode).
- server: Launches the server mode.

There are also some general options:
- -h, --help: Shows this screen.
- -v, --version: Shows application version number.
- -l, --language <lang_id>: Change the application language.

Every command has its associated options and parameters, explained in the subsections below, and its own help function, so in order to see the available options for the “check” command you can “run dpf_manager check --help”.
3.1.1 Check

Usage: dpf-manager check [options] source1 [source2 ... sourceN]

Where sources is the tiff’s list to scan. Each element of the list can be either:

- A file. It can be an absolute or a relative path. Example: "image.tif"
- A folder. All the files inside the folder will be processed. Example: "photos/tifimages"
- An URL. Example: "http://www.dpfmanager.org/image1.tif"
- A zip file. Example: "images.zip"

The following options are available:

- `-c, --configuration <filename>`: Selects a configuration file.
- `-f, --format [xml, json, pdf, html]`: Specifies the report format (overriding the one in the config file). Default is 'xml,html'.
- `-h, --help`: Shows this screen
- `-o, --output <path>`: Specifies the output folder (overriding the one specified in the config file).
- `-r, --recursive [depth]`: Check directories recursively. If nodepth is given (-r) then it is fully-recursive. Default is '-r 1'.
- `-s, --silence`: Silent execution (do not display info messages in the console)
- `--show-report`: Open the report at the end
- `-t, --threads <N>`: Specify maximum number of threads used for checking. By default, SO chooses.
- `-w, --overwrite`: Overwrites the output folder
- `-q, --quick`: Performs a quick check

3.1.2 Config

Usage: dpf-manager config [action] [options]

Actions (only one action allowed):

- `-a, --add <name>`: Creates a new configuration. The name can be either a path or a configuration file name (it will be places in the default configuration folder).
- `-e, --edit <name>`: Edits a configuration. The name can be either a path or a configuration file name (placed in the configuration folder).
- `-h, --help`: Shows this help.
- `-i, --info <name>`: Shows a configuration. The name can be either a path or a configuration file name (placed in the configuration folder).
- `-l, --list [type]`: Lists useful information depending on 'type'. Type can be:
  - 'iso': Lists all possible ISOs.
  - 'rule': Lists all accepted tags for rules.
  - 'fix': Lists all accepted tags for fixes.
  - 'autofix': Lists all autofixes.
  - none: Lists all configurations in default folder.
-r, --remove <name>: Deletes a configuration. The name can be either a path or a configuration file name (placed in the configuration folder).

Options:
- --autofix <autofix>: Adds an autofix. Use 'dpf-manager config --list autofix' to see the autofixes list.
- -d, --description <description>: Sets the description of the configuration. If description is 'EMPTY', it will remove the description.
- --disable-iso-rule <iso_id> <rule_id>: Disables a rule of an ISO. Use 'dpf-manager config --list iso' to see the ISOs list.
- --enable-iso-rule <iso_id> <rule_id>: Enables a rule of an ISO. Use 'dpf-manager config --list iso' to see the ISOs list.
- --fix <operator> <tag> [value]: Adds a fix specified by the parameters. See fix specification below.
- -f, --format '[xml, json, pdf, html]': Specifies the report format. Default is 'xml,html' when creating.
- --iso <iso_id>: Adds a ISO to check. Use 'dpf-manager config --list iso' to see the ISOs list.
- -o, --output <path>: Specifies the output folder. Path can be either a path or 'DEFAULT', which will set the output folder to the default one.
- --remove-autofix <autofix>: Removes an autofix.
- --remove-fix <operator> <tag> [value]: Removes a fix specified by the parameters. See fix specification below.
- --remove-iso <iso_id>: Removes a ISO to check.
- --remove-rule <type> <tag> <operator> <value>: Removes a rule specified by the parameters. See rule specification below.
- --rule <type> <tag> <operator> <value>: Adds a rule specified by the parameters. See rule specification below.

Rule specification:
- Type must be 'error' or 'warning'.
- Tag must be an accepted Tag. Use 'dpf-manager config --list rule' to see the list of accepted tags.
- Operator must be 'GT' (Greater than), 'LT' (Less than) or 'EQ' (Equals).

Example: --rule error ImageWidth GT 500
Example: --rule warning ImageLength EQ 500

Fix specification:
- Operator must be 'addTag' or 'removeTag'.
- Tag must be an accepted Tag. Use 'dpf-manager config --list fix' to see the list of accepted tags.
- Value is the value of the added tags.

Example: --fix removeTag Copyright
Example: --fix addTag Artist John
3.1.3 Modules

Usage: dpf-manager modules [options]

Options:
- `-a, --add <name> <path>`: Add a conformance checker.
- `--configure <name> <configuration>`: Set the configuration of a conformance checker.
- `--disable <name>`: Deactivate a conformance checker.
- `-e, --edit <name> <path>`: Edit the execution path of a conformance checker.
- `--enable <name>`: Activate a conformance checker.
- `--extensions <name> [ext1,ext2..extN]`: Set the extensions of a conformance checker. For example ‘tif,tiff’.
- `-h, --help`: Shows this screen.
- `-i, --info <name>`: Get information of a conformance checker.
- `-l, --list`: Show the list of conformance checkers.
- `--parameters <name> <parameters>`: Set the input parameters of a conformance checker.
- `-r, --remove <name>`: Remove a conformance checker from the list.

3.1.4 Periodic

Usage: dpf-manager periodic [options] [source1 ... sourceN]

Options:
- `-a, --add`: Add a new periodical check. It requires the options `--periodicity`, `--configure` and the sources.
- `--configure <filename>`: Specifies the configuration file.
- `-e, --edit <id>`: Edits the periodical check with the specified id. It requires the options `--periodicity`, `--configure` and the sources.
- `-h, --help`: Shows this screen.
- `-l, --list`: Lists all periodical checks.
- `--periodicity <D|W|M> [extra_info]`: Specifies the periodical check mode. It must be D (daily), W (weekly) or M (monthly). In case of weekly, extra_info must be a list of the week days (with numbers, where 1 = Monday and 7 = Sunday) separated by ‘,’. In case of monthly, extra_info must be the number of the day of the month (between 1 and 28, both included).
- `-r, --remove <id>`: Removes the periodical check with the specified id.
- `--time <HH:mm>`: Specifies the time for the periodical check. The time format must be HH:mm. Default value is '00:00'.

3.1.5 Remote

Usage: dpf-manager remote [options] [source1 ... sourceN]
(the sources can be single files, directories, zip files or URLs)

Options:

- `-c`, `--configuration <filename>`: Selects a configuration file.
- `-f`, `--format [xml, json, pdf, html]`: Specifies the report format (overriding the one in the config file). Default is 'xml,html'.
- `-h`, `--help`: Shows this screen
- `-j`, `--job <job_id>`: Get job state.
- `-o`, `--output <path>`: Specifies the output folder (overriding the one specified in the config file).
- `-u`, `--url <url>`: Specifies the remote server url. For example "http://example.com/dpfmanager", and also with a custom port "example.com:9000/dpfmanager".
- `-w`, `--wait`: Wait for a remote check to finish. Default is false.

3.1.6 Server

Usage: dpf-manager server [options]

Options:

- `-h`, `--help`: Shows this screen
- `-p`, `--port <port_number>`: Specifies a port number. Default port is randomly chosen.

3.1.7 Results of the execution

The results of the execution can be found in the specified output folder, or the default user data directory (in Windows: “c:\users\YOUR_USER\DPF Manager”, in Linux and MacOS: “/home/YOUR_USER”) if no output folder has been specified. Each one of the analysed tiff images will have its own report (in the specified formats), and a summary report will be also generated.

In case that some fixes have been defined in the configuration file, the fixed images will be stored in the “fixed” directory, and each image will have generated an additional report for the fixed image, where the differences with the original image will be highlighted.

3.2 Client-Server mode

The CLI can be used for client/server mode using the commands server to create the server instance and remote to use the client mode.

When running the conformance checker in client mode, it does not halt until the job is done, but a job id is returned instead, which can be used to know the state of the job at any time through a -job call. The next call after the job is finished will return also the report. The option `-w` changes this behaviour allowing the client to wait until the job is done, instead of returning its job id.
3.3 Graphical User Interface

The main window of the graphical user interface (GUI) is the following:

The input file (or folder, or url, or zip) can be defined by:

- manually typing its location
- dragging the input into the text box
- clicking the “Select” button in the right side (the arrow allows specifying files or folders to select).

In order to verify the files, a configuration needs to be defined. The software comes with a default configuration file, which verifies that the files conform to the Tiff Baseline 6.0 and generates a report in HTML format.
New configuration files can be defined by clicking the “New” button. Configuration files can also be edited, removed or imported from other sources than the default.

Once both the files to be analysed, and the configuration file have been defined, the check buttons starts the process. A full check validates the tiff files and shows all the errors found in the report, while a quick check only shows if the tiff files have any error or not.

The language of the GUI can be changed using the bottom-right selector.

**3.3.1 Create Configuration**

The creation of a new configuration file consists in 5 steps.

In the first step, the ISO standards to be checked are selected.

As mentioned in the above note, the DPF Manager automatically checks the validity of all the available ISOS, and identifies in the report the standards where it conforms to, even if they have not been selected in this step.
In the second step the policy checker is defined. It consists in two parts. In the first part (custom standards), each of the standards selected in the implementation checker step can be edited. In the second part the acceptance criteria of the organization can be defined by adding rules with the desired properties for the Tiff files.

When editing an ISO a new panel will appear showing all of the rules that the iso validates for a file to be valid. These rules can be disabled in order to create a less restrictive implementation check.
In the second part rules can be added with the desired properties for the Tiff files.

Rules can be **mandatory**, meaning that images not satisfying the rule will produce a policy error, or can be **warnings**, meaning that the images satisfying the rule will produce a warning.

The available policies are the following:

- **ImageWidth**: Checks the width in pixels of the image. This corresponds to tag 256 ImageWidth.
- **ImageLength**: Checks the height in pixels of the image. This corresponds to tag 257 ImageWidth.
- **PixelDensity**: Checks the resolution, in pixels per centimetre. This corresponds to tags 282 XResolution and 283 YResolution, which have usually the same value.
- **NumberImages**: Checks the number of images in a single Tiff file. This is the number of IFD in the TIFF file.
- **BitDepth**: Number of bits per pixels component (multiples of 2). This corresponds to tag 258 BitsPerSample.
- **ExtraChannels**: Number of extra pixel components (e.g. transparency).
- **EqualXYResolution**: Checks that the X and Y resolution of the image are the same.
- **Compression**: Compression scheme. This corresponds to tag 259 Compression.
- **Photometric**: Color space of the image data. This corresponds to tag 262 PhotometricInterpretation.
- **Planar**: How the pixels components are stored. This corresponds to tag 284 PlanarConfiguration.
- **ByteOrder**: Byte order (little endian, big endian).
- **FileSize**: The size of the file in bytes.
- **ICCProfileClass**: Class of the device ICC Profile.

In the third step, the format of the report and the output folder is chosen:

![Output Format Options](image)

By default the reports are stored in the user data directory (in windows operating systems it is “c:\users\USER_NAME\DPF Manager”, in Linux and MacOs it is “/home/USER_NAME/DPF Manager”). But a custom folder can be also set.

The step number four is the metadata fixer, where the user can specify which fixes have to be done in the image. A fix means that a new image will be generated with the changes specified in this section, for example, adding or removing metadata into the image.
There are four autofixes available:

- **Clear Private Data**: This removes all the information related with the GPS coordinates where the photo was taken.
- **Fix non-Ascii tags**: This solves a common error related with text that is not encoded in 7-bit ascii, which is the encoding allowed for tiff.
- **Make Baseline Compliant**: Fixes some common errors regarding the Baseline 6 specifications.
- **Fix Metadata Inconsistencies**: Resolves incoherencies in the metadata coming from IPTC, XMP and EXIF, following the guidelines of the Metadata Working Group\(^1\).

\(^1\) [http://www.metadataworkinggroup.org/specs/](http://www.metadataworkinggroup.org/specs/)
Finally, a summary window shows all the configuration properties for revision. And the whole configuration can be saved into a file, providing its name and a short description (optional).

3.3.2 Editing a configuration

Configurations can be also opened and modified by clicking the button “Edit” in the main window. The steps of the wizard are going to be the same, and finally the configuration will be saved either to the same file or to a new one.

3.3.3 Importing a configuration

External configuration files can also be imported to the configuration files folder through the “Import” button in the main window. When a configuration file is imported, the program asks if you want to save it in the default configuration directory, so that in future executions the configuration will appear in the configurations list, or not.
3.3.4 Periodical checks

The periodical checks tab allows the user to define checks to be performed periodically.

In order to configure a periodical check, a source (file or folder) must be defined and a configuration file has to be selected. Then, the periodicity can be either daily, weekly or monthly, and we can specify also the desired time of the check.

Periodical checks create tasks in the operating system, so it is not necessary for the DPF Manager to be left opened for the periodical checks to be run.
3.3.5 Conformance checkers

External conformance checkers can be configured in the “Conformance checkers” tab.

By default, only the built-in TIFF conformance checker is available, but we can define other conformance checkers for other file formats (file extensions). Then we have to specify the location of the external conformance checker, the arguments to run it, and its configuration file.

Note that you can define more than one conformance checker for the same file extension. In this case, the DPF Manager will automatically leverage the load over all the available conformance checkers of the same format.
3.3.6 Tasks

A tasks widget is available in the bottom of the window. By default it is minimized, but clicking in the “Tasks” button in the bottom-left will show it.

Here, all the checks that have been done in the current execution, or are currently running, can be seen. The icon in the left shows the main report format for the completed tasks, and clicking on it will show the report.

When a task is running, it can be paused, resumed and cancelled.

3.3.7 Console

A console widget is also available, which is used for the application as a logging system, and to show to the user possible errors that could occur during the execution.
3.3.8 Report Window

After the report has been generated, the icon in the task takes the user to the report window, where a summary of the validation is shown.

At the top, there is a table with a pie graph showing the total number of files analysed, and the total number of errors, correct files, and valid files with warnings. At the right side, the different formats for the global report are shown and can be clicked to create the corresponding file.

The table at the bottom shows all the files analysed and the errors and warnings found. Also, a list of formats is available in order to generate the individual reports.

When quick checks are performed instead of full checks, the errors and warnings columns are not shown, but an option to generate full checks of the individual reports is available, as well as a button to generate full checks of the entire set of files.
An HTML-formatted global report looks like this, similar to the information shown before.
The individual reports show the details of the Tiff structure, including a tree object, the metadata inconsistencies, the tag list and the results of the validation of each one of the selected profiles.

In case the Tiff file contains multiple images, they can be chosen from the File Structure section, and the Tags list will be updated.

For JSON and XML reports, the tag list contains all the existing tags in the Tiff file, while in human-readable formats, i.e. PDF and HTML, only the most relevant tags are displayed.
3.3.9 Summary of reports window

The summary reports window shows a table of all the reports generated with the DPF Manager. Each row shows the most relevant statistics of the report, and the formats in which the report was generated.

By clicking in the formats icons, the reports can be examined.

The reports folder can be cleaned using the “clear options” button in the bottom left, which allows to remove all the reports, or only reports older than an specific date.
3.3.10 Statistics module

The statistics module shows an overview of all the reports that have been performed with DPF Manager. In the top table, the total number of validations is shown, with basic general information.

The second table shows all the tags that have been found in the reports, sorted by frequency. The information is divided in main images and thumbnail images (TIFF files can include reduced versions of a full resolution image). Each tag can be clicked in order to show the different values that have been found in the analysed reports.

After that, the implementation checker table, shows the different ISOSs that have been analysed, and on clicking in them, an additional table appears showing all the errors that have been encountered in the files analysed with this standard.
Finally, the policy checker table, shows all the policies that have been defined in any report and the results that have been obtained.
3.3.11 First time run

The first time the application is opened, a legal advice is displayed.

Here the user can send their contact information and tell the application whether to send anonymous usage information to the development team.

The feedback option can be later modified in the About tab.