



Octopus

User Manual



HORIZON 2020

sdn-userdesk@seadatanet.org – www.seadatanet.org

SeaDataNet - The pan-European infrastructure for marine and ocean data management

| | |
|---|---------------------|
| Deliverable number | Short title |
| | OCTOPUS user Manual |
| Long title | |
| OCTOPUS user Manual | |
| Short description | |
| <p>Octopus is a multi-formats splitter & converter tool. It replaces the following software: med2medSDN, Change_vocab_V1toV2, MedSDN2CFPoint, OdvSDN2CFPoint, offering a unique and ergonomic tool.</p> <p>It also allows :</p> <ul style="list-style-type: none"> • to split a multistation file into monostation ones • to extract 1 to n stations from a multistation file and export them into another multistation file or several monostation ones. • to check the compliancy of MedAtlas, ODV and ODV variant formats, netCDF-CFPoint and netCDF for HF-Radar | |
| Author | Working group |
| S. Brégent, M. Fichaut, S. Crouzille | |
| Dissemination | Copyright terms |
| Public | |

History

| Version | Authors | Date | Comments |
|---------|------------------------|------------|--|
| 1.0 | S. Brégent, M. Fichaut | 07/01/2016 | Creation |
| 1.1 | S. Brégent, M. Fichaut | 19/12/2016 | Add installation instructions. Change export types names from “mono” to “split” and from “multi” to “keep” 5.2. Specifications for MGD: 81 and 98 are different formats. They cannot be converted from the same input directory. |
| 1.2 | S. Brégent, M. Fichaut | 27/01/2017 | Add warning on MGD formats: MGD81 and MGD98 are distinct formats (§5.1). Add warning on output path in case of directories: do not write the last separator (§6) |
| 1.3 | S. Brégent, M. Fichaut | 21/07/2017 | <ul style="list-style-type: none"> • Add information on BODC vocabularies checks (§4.2) • Output file type choice: modification of the label in the graphical interface (no modification for batch mode) (§5.3 and §5.4) |



| | | | |
|-------|-----------------------------|------------|--|
| | | | <ul style="list-style-type: none"> • Additional information about log files (§5.6) |
| 1.4 | S. Brégent, M. Fichaut | 04/03/2019 | <ul style="list-style-type: none"> • Add procedure to launch octopus behind a proxy (§3.1, §3.2) • Add a procedure to launch in batch mode, on a server without a graphical interface (§6) • add option for check only in batch mode (§6) • Modify Octopus requirements and installation • Add a procedure to update external resources in batch mode |
| 1.5 | S. Brégent, M. Fichaut | 05/06/2019 | Take into account ODV variants for biology, microlitter and flowcytometry for checks and conversion ODV to ODV. |
| 1.5.3 | S. Brégent, M. Fichaut | 24/03/2020 | <ul style="list-style-type: none"> • For all conversions/split cases, update URLs for CDIs, CSRs and Nerc vocabularies (§4.2) |
| 1.6.0 | S. Crouzille, M. Fichaut | 07/01/2021 | <ul style="list-style-type: none"> • Add a table of file formats that can be checked. • Add “HF Radar NetCDF” as a format file that can be checked. |
| 1.6.1 | S. Crouzille | 11/03/2021 | <ul style="list-style-type: none"> • Add HTTPS elements to the launch command through a proxy. |

Table of contents

| | |
|---|----|
| 1. Introduction | 5 |
| 2. Requirements | 5 |
| 3. Installation..... | 5 |
| 3.1. Launch on windows..... | 6 |
| 3.2. Launch on Linux..... | 6 |
| 4. Get started | 7 |
| 4.1. Settings..... | 7 |
| 4.1.1. Menu Edit/settings..... | 7 |
| 4.1.2. Menu Edit/Coupling table | 8 |
| 4.2. Features..... | 9 |
| 4.2.1. File checks | 9 |
| 4.2.2. File conversions/splits | 9 |
| 5. Use of OCTOPUS in interactive mode | 11 |
| 5.1. Open input file or directory..... | 11 |
| 5.2. Check input file(s)..... | 12 |
| 5.3. Split to mono station files | 12 |
| 5.4. Select the output file or directory | 12 |
| 5.5. Select the LOCAL CDI ID(s) for ODV file(s) | 12 |
| 5.6. OCTOPUS log file | 13 |
| 6. Use of OCTOPUS in batch mode..... | 14 |

1. Introduction

Octopus is a multi-formats checker, converter and splitter tool. It replaces the following software: med2medSDN, Change_vocab_V1toV2, MedSDN2CFPoint, OdvSDN2CFPoint, offering a unique and ergonomic tool.

- OCTOPUS checks the compliance of a file to the SeaDataNet **ODV**, **ODV variants** for biology, flow cytometry and microlitter, **netCDF** (CFPOINT), **netCDF for HF-Radar** and **MedAtlas** standard formats.
- OCTOPUS converts files in a given SeaDataNet format to another SeaDataNet format (e.g.: ODV to netCDF, netCDF to ODV, MedAtlas to NetCDF, MedAtlas to ODV)
- OCTOPUS has also additional functions such as:
 - Split a multi-station SeaDataNet file into mono-station SeaDataNet files
 - Extract station(s) from SeaDataNet files
 - Convert MGD v81 and v98 to SeaDataNet ODV files

Octopus can be used in interactive mode or in batch mode.

2. Requirements

Octopus is written in Java.

Octopus is available in 2 versions:

- a "classic" version, without Java. It requires java 1.8.0_131 or greater (but lower than 1.9) already installed in the system.
- a "standalone" version, with Java 1.8.0_201 bundled into it (located in the `octopus/jre` directory).

This version doesn't require Java. Octopus is available for multiple platforms: Windows, Linux.

32 bits platforms are not supported.

Languages: French, English

3. Installation

Download the Octopus software from SeaDataNet web site, under Standards & Software:

<http://www.seadatanet.org/Standards-Software>

Simply choose the version according to your system (Linux/Windows), and your Java configuration ("classic", "standalone").

Copy the zip file on your computer and unzip it.

The change log is available in octopus menu help> about



sdn-userdesk@seadatanet.org – www.seadatanet.org

SeaDataNet - The pan-European infrastructure for marine and ocean data management

3.1. Launch on windows

To launch Octopus, double click on octopus.exe in the octopus directory.

If your internet connection has a proxy server, please use the following procedure:

1. In the octopus installation directory (containing the octopus.jar file): create a file named octopus.bat
2. in this file, write the following line:

```
[directory\]java -D http.proxyHost=xx.xx.xx.xx -D http.proxyPort=yy -jar octopus.jar
```

where

- xx.xx.xx.xx = your proxy IP address
- yy = your proxy port
- [directory\] is the optional `java.exe` location, depending on your Java installation.

In the "standalone" version, the directory is `.\jre\bin\`. If you already have Java installed, no directory is needed.

In case of a HTTPS proxy, you may use similar options:

```
[directory\]java -D https.proxyHost=xx.xx.xx.xx -D https.proxyPort=yy -jar octopus.jar
```

HTTP and HTTPS proxy parameters can be combined if needed:

```
[directory\]java -D http.proxyHost=xx.xx.xx.xx -D http.proxyPort=yy -D https.proxyHost=aa.aa.aa.aa -D https.proxyPort=bb -jar octopus.jar
```

3. launch Octopus using this octopus.bat file (double click) instead of the octopus.exe file.

3.2. Launch on Linux

To launch Octopus, double click on octopus.sh in the octopus directory, or launch it from a terminal:

```
cd octopus
```

```
./octopus.sh
```

If your internet connection has a proxy server, please use the following procedure:

1. In the octopus installation directory (containing the octopus.jar file): create a file named octopus_proxy.sh
2. in this file, write the following line:

```
[directory/]java -D http.proxyHost=xx.xx.xx.xx -D http.proxyPort=yy -jar octopus.jar
```

where

- xx.xx.xx.xx = your proxy IP address
- yy = your proxy port
- [directory/] is the optional `java` binary location, depending on your Java installation.



In the "standalone" version, the directory is `./jre/bin/`. If you already have Java installed, no directory is needed.

In case of a HTTPS proxy, you may use similar options:

```
[directory/]java -D https.proxyHost=xx.xx.xx.xx -D https.proxyPort=yy -jar octopus.jar
```

HTTP and HTTPS proxy parameters can be combined if needed:

```
[directory/]java -D http.proxyHost=xx.xx.xx.xx -D http.proxyPort=yy -D https.proxyHost=aa.aa.aa.aa -D https.proxyPort=bb -jar octopus.jar
```

3. launch Octopus using this `octopus_proxy.sh` file (double click)

4. Get started

4.1. Settings

4.1.1. Menu Edit/settings

This item is used to configure OCTOPUS for your own utilization.

You can:

- Choose your language.
- Choose your default input and output directories. The browse button will automatically open these directories.
- Choose your EDMO code (used to convert MGD files and non SDN MedAtlas file to SDN files).
- Ask Octopus to fill the local coupling table, by checking "Use coupling table" and choosing the coupling prefix (path prefix that will NOT be written in the coupling files path).
- Update external lists (EDMO codes and BODC NERC Vocabularies).



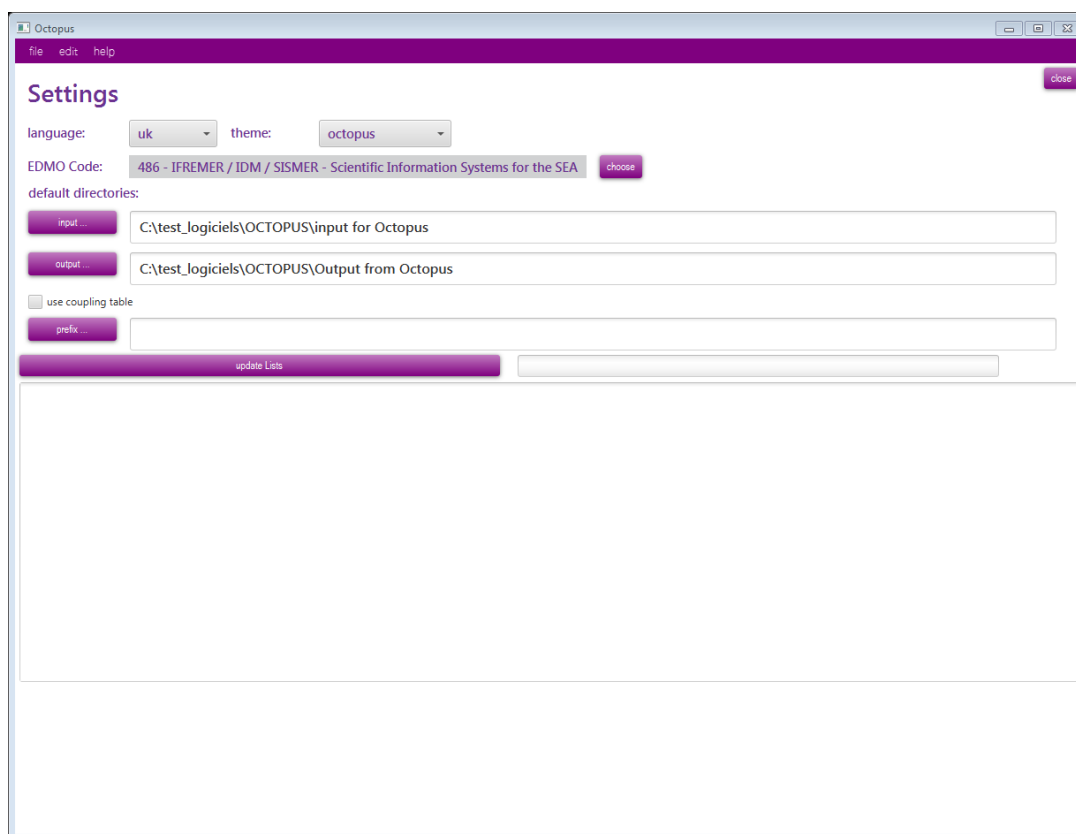


Figure 1 – OCTOPUS settings windows

4.1.2. Menu Edit/Coupling table

The “Coupling Table” menu of OCTOPUS allows basic management of the content of the coupling table used by the SeaDataNet download manager to retrieve the LOCAL_CDI_IDs requested by a user downloading.

4.1.2.1. Export

This export function generates a coupling file that will be used by SeaDataNet Download Manager for retrieving stations in the files.

The coupling file used by SeaDataNet download manager is unique and called “coupling.txt”. This file must not contain duplicates (the coupling file ID is LOCAL_CDI_ID + format) It’s up to OCTOPUS user to create only one file called “coupling.txt” for SeaDataNet purpose by using the coupling table facilities.

The format of this coupling file is the following:

LOCAL_CDI_ID;Modus;Format;File_name

The export function will replace the previous coupling file if the name of the export file is the same.

4.1.2.2. Delete all

The delete all button is used to empty the coupling table. All records will be deleted. The table should be first exported if the user wants to keep the information in a flat file.

4.2. Features

4.2.1. File checks

Available file checks are listed in the table below.

Table 1 - Possible format checks using OCTOPUS (ODV variants stands for biology, microlitter and flow cytometry data)

| Format | Check |
|-----------------------|-------|
| MedAtlas non SDN | ✗ |
| MedAtlas SDN | ✓ |
| ODV SDN | ✓ |
| ODV variants | ✓ |
| netCDF – SDN CFPoint | ✓ |
| HF-Radar (SDN netCDF) | ✓ |
| MGDv81 | ✗ |
| MGDv98 | ✗ |

4.2.2. File conversions/splits

Available file conversions/splits are listed in the table below

Table 2 - Possible format conversions using OCTOPUS (ODV variants stands for biology, microlitter and flow cytometry data)

| output→ input ↓ | MedSDN | ODV | ODV variants | netCDF -CFPoint |
|--------------------|--------|-----|--------------|-----------------|
| Med non SDN | ✓ | ✓ | ✗ | ✓ |
| Med SDN | ✓ | ✓ | ✗ | ✓ |
| ODV SDN | ✗ | ✓ | ✗ | ✓ |
| ODV variants | ✗ | ✗ | ✓ | ✗ |
| netCDF-CFPoint | ✗ | ✓ | ✗ | ✓ |
| MGDv81 | ✗ | ✓ | ✗ | ✗ |
| MGDv98 | ✗ | ✓ | ✗ | ✗ |

Input can be a file, or a directory containing several files (all files using the same format, no sub-directory).

Automatic format updates:

- For all conversion/split cases, Octopus will automatically add SDN CDI references.

- For conversion/split from MedAtlas (SDN or non SDN) to MedAtlas SDN, existing SDN CSR and SHIP (NVS2CON) references will be added to output file(s).
- For conversion/split from MedAtlas non SDN to MedAtlas SDN, SDN mapping lines will be added to output file(s).
- For all conversion/split cases, Octopus will automatically check the BODC vocabularies terms. Octopus will automatically replace deprecated term, and display an error if term does not exists or is deprecated and has no “ReplacedBy” attribute.
- For all conversions/split cases, update URLs for CDIs, CSRs and NERC vocabularies

5. Use of OCTOPUS in interactive mode

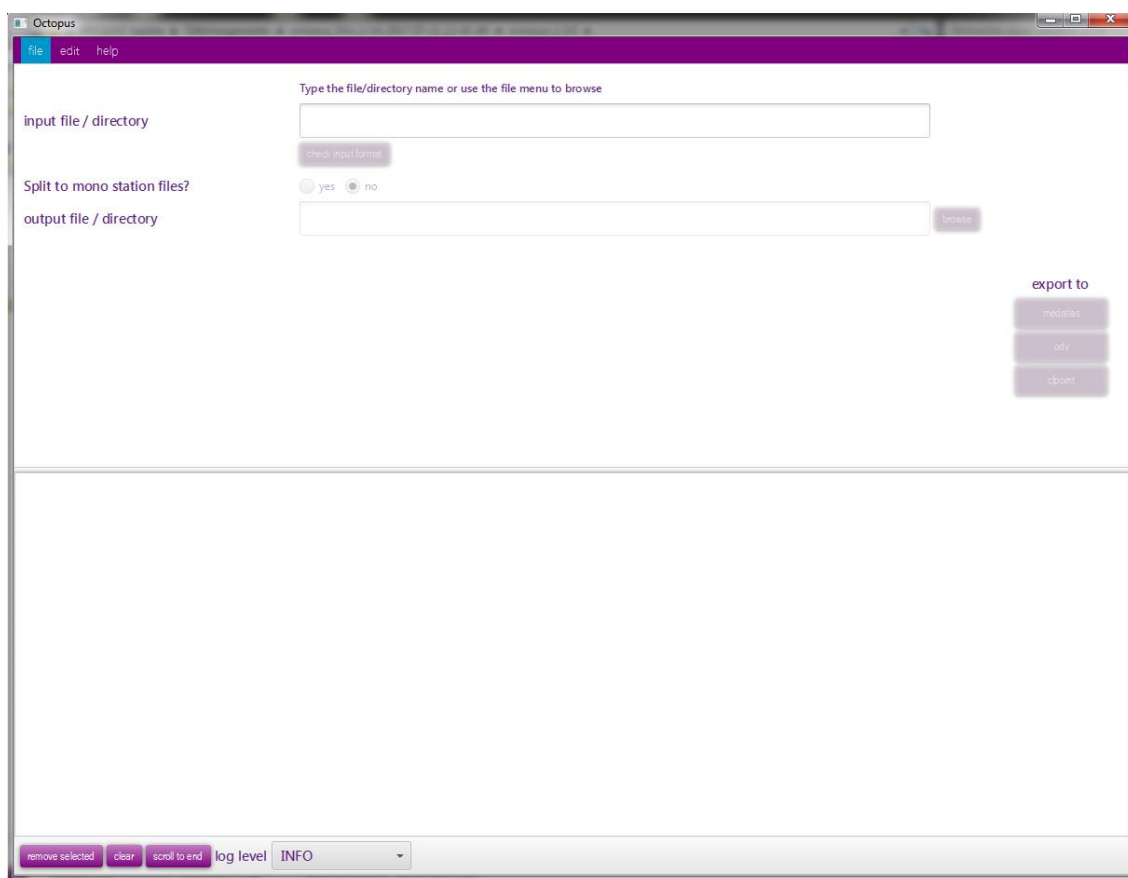


Figure 2 – Main screen of OCTOPUS

5.1. Open input file or directory

This step allows to select the file(s) to be converted/split.

OCTOPUS is able to process one file or one directory containing files at the same format.

WARNING: MGD81 and MGD98 are not the same format. They must be gathered in separated directories.

There are two possible ways to open a file:

- Menu file/open file
- Write or paste the input path in the input file/directory then TAB or ENTER

There are two possible ways to open a directory:

- Menu file/open directory
- Write or paste the input path in the input file/directory then TAB or ENTER

5.2. Check input file(s)

Once the file or directory has been chosen, it is possible to check the format of the file(s) by clicking on the “Check the input format” button. In the case of a directory, all files are supposed to be at the same format (it is not possible to check a directory containing MedAtlas and ODV files, or netCDF-CFPoint and ODV files, for example).

5.3. Split to mono station files

This feature is not available for MGD files.

If the user clicks on Yes, the multi station input file will be split to n mono station files.

By default No is selected.

5.4. Select the output file or directory

Use the browse button, or write or paste the output path in the output file/directory.

If input is a file, and output type is multi, you have to input a filename.

Otherwise, set a directory name.

Note on output files names:

Except the case of one input file exported as a multi-stations file, Octopus will generate paths as described below.

| | | Split to mono station files | |
|------------|-----------|--|--|
| | | Yes (interactive mode) = Split (batch mode) | No (interactive mode) = Keep (batch mode) |
| Input type | File | output/LOCAL_CDI_ID.ext | output |
| | Directory | output/inputFileName/ LOCAL_CDI_ID.ext | output/inputFileName.ext |

where : - directories are in **red**, files are in **green**.

- Output is the path set in the output file/directory field
- inputFileName is the name of one input file in the input directory
- .ext is the extension of the filename: .txt for ODV, .nc for netCDF-CFPOINT and .<free text> for MedAtlas

5.5. Select the LOCAL CDI ID(s) for ODV file(s)

This field is available only for MGD input files or directories. MGD files do not have LOCAL CDI IDs. You have to specify it.

In case of an input file, write the LOCAL CDI ID in the field.

In case of an input directory, you will have to write a mapping file. Use the Browse button to select the mapping file path, or write or paste the output path in the field.

Write a mapping file:

The mapping file is a semi-colon separated file, with two columns: file name and LOCAL CDI ID

Example:

20002001ATE.mgd77;FI29_2002AT

20003001ATE.mgd77;FI29_2003AT

5.6. OCTOPUS log file

While checking and/or converting OCTOPUS logs information, warnings and errors in the lower part of the main window and in the log file *octopus.log* located under *[octopus_install_folder]/logs*.

In the main octopus log window: information are written in black, warnings are written in orange and Errors are written in red.

If the check/conversion is OK, a green message is also written in the window.

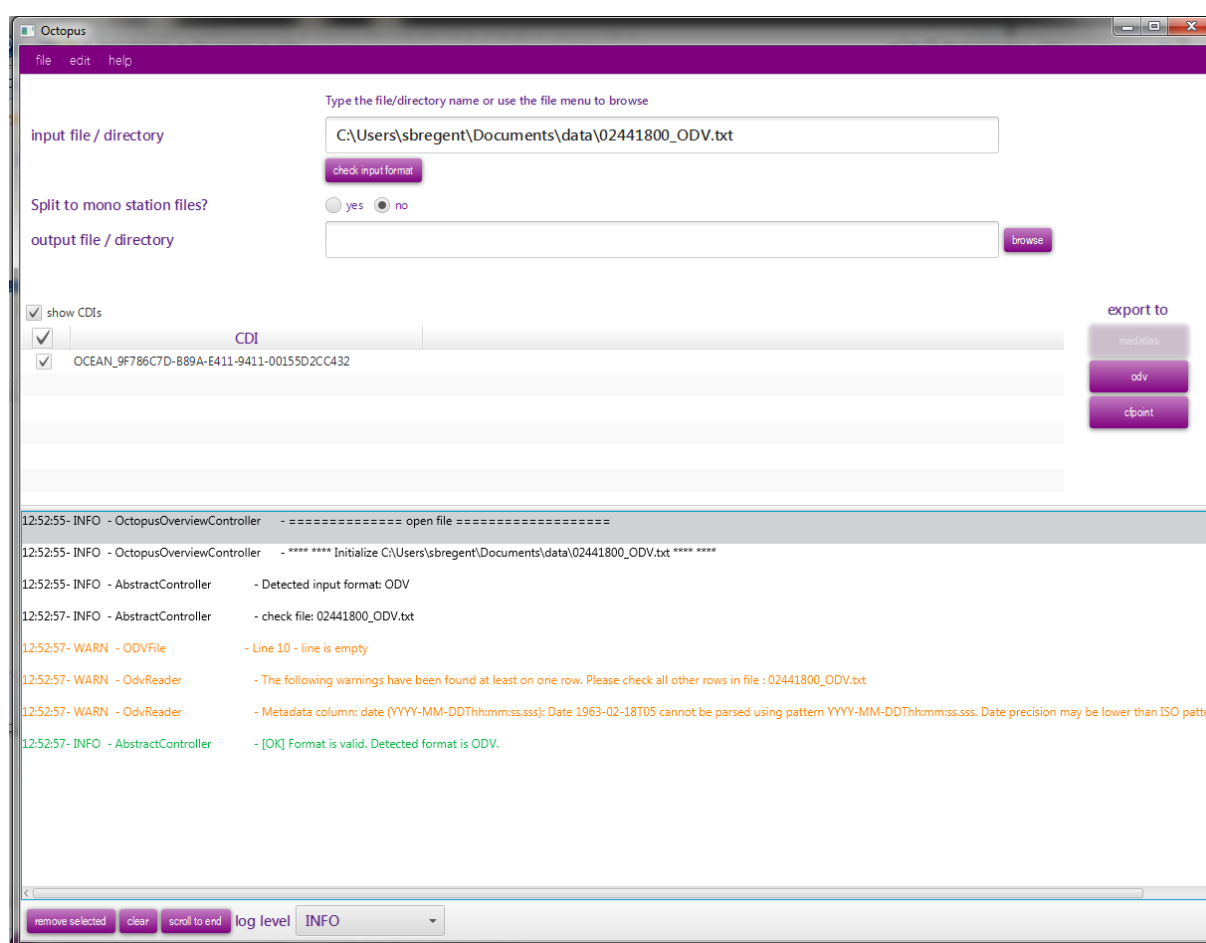


Figure 3 – Log information in OCTOPUS main screen

The max size of the log file is limited to 20 Mb. If the size exceed 20Mb, Octopus will create a zip file octopus-[YYYY-MM-DD]-[number].log.zip with this 20 Mb of information and open a new current octopus.log file.

6. Use of OCTOPUS in batch mode

Open a console and move to the Octopus installation directory (where the octopus.jar is)

Launch command: `java -jar octopus.jar <options>`

If you need to launch octopus on a server without a graphical interface (no graphical packages installed), use the command below:

```
java -cp octopus.jar fr.ifremer.octopus.controller.BatchController <options>
```

Usage:

Check a file or a directory

[directory\] `java -jar octopus.jar -i <arg> -check`

Convert a file or a directory

[directory\] `java -jar octopus.jar [-c <arg>] -f <arg> -i <arg> [-l <arg>] -o <arg> [-t <arg>]`

Update external resources

[directory\] `java -jar octopus.jar -update`

where [directory\] is the optional java binary location, depending on your Java installation:

In the "standalone" version, the directory is `.\jre\bin\`. If you already have Java installed, no directory is needed.

In [octopus_install_folder]/logs, Octopus in batch mode generates also a log file using JSON format, which can be easily read by a software.

| Argument | O/M | Comment |
|----------|----------------------------------|---|
| -check | Optional | check a file or a directory (no conversion) |
| -c <arg> | Optional | list of LOCAL_CDI_IDs, eg <FI35AAB, FI35AAC>, all CDIs are exported if this argument is omitted |
| -f <arg> | Mandatory | output format: <medatlas>, <odv> or <cfpoint> |
| -i <arg> | Mandatory | input path: </home/user/...> |
| -l <arg> | Mandatory if input is MGD | LOCAL_CDI_ID value if input is a file, mapping file if input is a directory (see §5.5) |
| -o <arg> | Mandatory | output path (file or directory): </home/user/...> |
| -t <arg> | Mandatory except if input is MGD | output type: <split> or <keep> |
| -update | optional | Update external resources (BODC vocabularies, CSR list, EDMO codes) |

WARNING:

If the output path is a directory, **DO NOT WRITE** a slash or backslash at the end of the path:

–o *"/home/out/exportCFDirectory"* is correct

–o *"/home/out/exportCFDirectory/"* is incorrect

Examples:

- Export all stations from input MedAtlas files directory to mono-stations CFPoint files

```
java -jar octopus.jar -i "/home/input/profileDir" -o "/home/out/exportCFDirectory" -f cfpoint -t split
```

- Export CDI1 and CDI2 stations from input MedAtlas file to multi-stations ODV file

```
java -jar octopus.jar -i "/home/input/profile.med" -o "/home/out/exportODV1and2.txt" -f odv -t keep -c CDI1,CDI2
```

- Export MGD file to multi-stations ODV file using XXX as local CDI ID

```
java -jar octopus.jar -i "/home/input/mgd81.mgd77" -o "/home/out/exportODV1and2.txt" -f odv -l XXX
```

While checking and/or converting OCTOPUS logs information in the log file *octopus.log* located under *[octopus_install_folder]/logs*.

The max size of the log file is limited to 20 Mb. If the size exceed 20Mb, Octopus will create a zip file *octopus-[YYYY-MM-DD]-[number].log.zip* file with this 20 Mb of information and open a new octopus.