

---

**InforION**

*Release 2.1*

**2021-05-18**



<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Installation</b>	<b>5</b>
2.1	Install from Package Managers . . . . .	5
2.2	Build from Source . . . . .	5
<b>3</b>	<b>Python Download for Windows</b>	<b>7</b>
3.1	Python 3 Installer Download . . . . .	7
3.2	Run the Installer . . . . .	8
<b>4</b>	<b>PIP</b>	<b>9</b>
4.1	Installation . . . . .	9
4.2	Upgrade . . . . .	9
4.3	Show Version . . . . .	9
<b>5</b>	<b>Jupyter Notebook to Windows 10</b>	<b>11</b>
<b>6</b>	<b>Generate M3 Excel Mapping File</b>	<b>15</b>
6.1	Description . . . . .	15
6.2	Parameters . . . . .	15
6.3	Example . . . . .	15
<b>7</b>	<b>Data Catalog CLI</b>	<b>17</b>
7.1	create . . . . .	17
7.2	delete . . . . .	18
<b>8</b>	<b>Data Lake CLI</b>	<b>19</b>
8.1	list . . . . .	19
8.2	get . . . . .	20
8.3	upload . . . . .	20
8.4	purge . . . . .	20
<b>9</b>	<b>Export LN Data</b>	<b>23</b>
9.1	Description . . . . .	23
9.2	Parameters . . . . .	23
9.3	Example . . . . .	24
<b>10</b>	<b>Transformation</b>	<b>25</b>

10.1	Description	25
10.2	Parameters	25
10.3	Example	25
<b>11</b>	<b>Loading</b>	<b>27</b>
11.1	Description	27
11.2	Parameters	28
11.3	Example	28
<b>12</b>	<b>Contributing guidelines</b>	<b>29</b>
12.1	Big changes	29
12.2	Code style	29
12.3	Tests	29
12.4	New Python dependencies	29
12.5	New non-Python dependencies	30
12.6	Style guide: Is it InforION or inforion?	30
12.7	Known ports/packagers	30
<b>13</b>	<b>Logging</b>	<b>31</b>
<b>14</b>	<b>Indices and tables</b>	<b>33</b>

InforION is a Python 3 application and library that interacts with InforOS. This library provides a series of functionalities to handle data via Infor ION:

- Migration into M3
- Import Data to/from the Infor Datalake
- Manage and load data from external systems into e.g. Infor M3 incl. data verification, logging reporting and workflow management via Infor ION.
- Export data from Infor LN/M3/EAM



# KAPITEL 1

---

## Introduction

---

InforION is a Python 3 application and library that interacts with InforOS

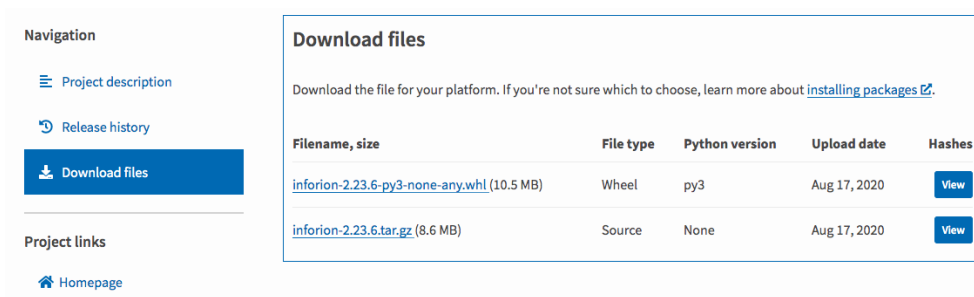




There are two ways that one can install Python3 Application: InforION. If you follow the steps below, you should have installed everything in no time.

### 2.1 Install from Package Managers

First of all, you can install InforION from *Package Managers*, which is available as a package on PyPI: <https://pypi.org/project/inforion/>.



The screenshot shows the PyPI page for the InforION package. On the left, there is a navigation sidebar with options: Project description, Release history, Download files (highlighted), and Project links (Homepage). The main content area is titled 'Download files' and includes a note: 'Download the file for your platform. If you're not sure which to choose, learn more about installing packages'. Below this is a table with columns: Filename, size, File type, Python version, Upload date, and Hashes. Two files are listed: 'inforion-2.23.6-py3-none-any.whl (10.5 MB)' and 'inforion-2.23.6.tar.gz (8.6 MB)'. Each row has a 'View' button next to the filename.

Filename, size	File type	Python version	Upload date	Hashes
<a href="#">inforion-2.23.6-py3-none-any.whl</a> (10.5 MB)	Wheel	py3	Aug 17, 2020	<a href="#">View</a>
<a href="#">inforion-2.23.6.tar.gz</a> (8.6 MB)	Source	None	Aug 17, 2020	<a href="#">View</a>

But, nevertheless, often the preferred way to install it is simply through **pip** in your *Terminal* (MacOS) or *Command Prompt* (Windows):

```
pip3 install inforion
```

### 2.2 Build from Source

Another easy way, is that Inforion may also be consumed and built directly from source. For this kind of installation we recommend the use of **Visual Studio Code** and opening its inside Terminal, so following these steps below wont be a problem.

```
git clone https://github.com/Fellow-Consulting-AG/inforion.git
```

As soon as you are inside os the new inforion directory, just run the following command:

```
make install
```

### 2.2.1 Install in 2 minutes

<https://asciinema.org/a/347875.svg>

---

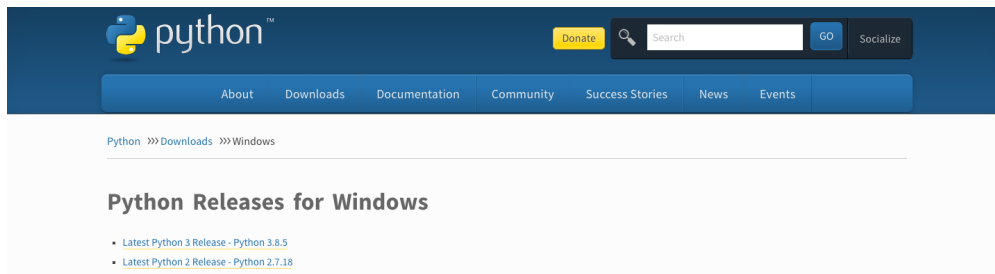
## Python Download for Windows

---

In this guide we are going to show you how to install **Python 3.7** or higher for your Windows.

### 3.1 Python 3 Installer Download

1. In order to install Python, you must download it from this URL, but it is very important that you select or use at least **Python 3.7 or higher**: [Python Releases for Windows](#)



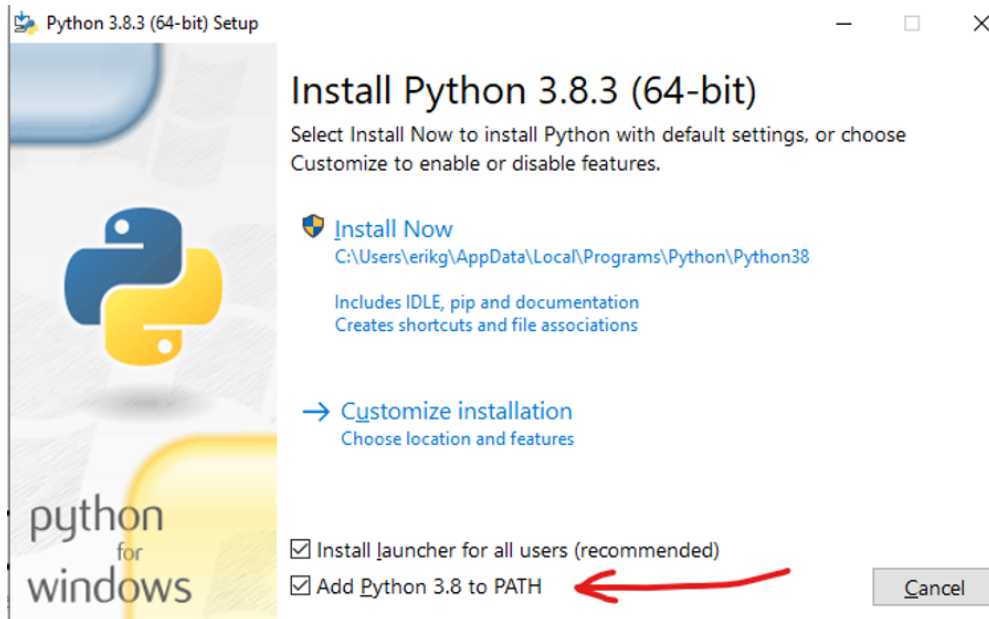
2. Then, scroll down and select **x86-64 executable installer** for Windows 10-64 bit computer or **Windows x86 executable installer** for 32-bit.

Version	Operating System	Description	MD5 Sum	File Size	PGP
<a href="#">Gzipped source tarball</a>	Source release		e2f52bcf531c8cc94732c0b6ff933ff0	24149103	<a href="#">SIG</a>
<a href="#">XZ compressed source tarball</a>	Source release		35b5a3d0254c1c59be9736373d429db7	18019640	<a href="#">SIG</a>
<a href="#">macOS 64-bit installer</a>	Mac OS X	for OS X 10.9 and later	2f8a736eeb307a27f1998cf07f22440	30238024	<a href="#">SIG</a>
<a href="#">Windows help file</a>	Windows		3079d9cf19ac09d7b3e5eb3fb05581c4	8528031	<a href="#">SIG</a>
<a href="#">Windows x86-64 embeddable zip file</a>	Windows	for AMD64/EM64T/x64	73bd7aab047b81f83e473efb5d5652a0	8168581	<a href="#">SIG</a>
<a href="#">Windows x86-64 executable installer</a>	Windows	for AMD64/EM64T/x64	0ba2e9ca29b719da6e0b81f7f33f08f6	27864320	<a href="#">SIG</a>
<a href="#">Windows x86-64 web-based installer</a>	Windows	for AMD64/EM64T/x64	eeab52a08398a009c90189248ff43dac	1364128	<a href="#">SIG</a>
<a href="#">Windows x86 embeddable zip file</a>	Windows		bc354669bffd81a4ca14f06817222e50	7305731	<a href="#">SIG</a>
<a href="#">Windows x86 executable installer</a>	Windows		959873b37b74c1508428596b7f9df151	26777232	<a href="#">SIG</a>
<a href="#">Windows x86 web-based installer</a>	Windows		c813e6671f334a269e669d913b1f9b0d	1328184	<a href="#">SIG</a>

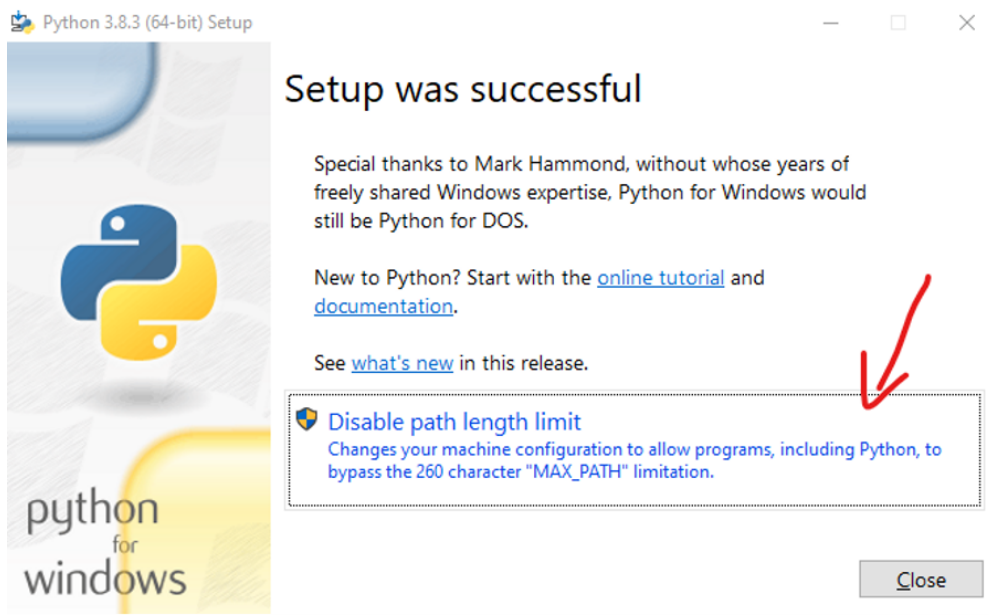
## 3.2 Run the Installer

Once you have downloaded an installer, simply run it by double-clicking on the downloaded file. A pop-up window should appear that looks like the following one:

3. Prior installing Python, please select the option: **Add Python 3.8 to PATH**



4. After Setup was successful select: **Disable path length limit**



Pip is a de facto standard package-management system used to install and manage software packages written in Python. Many packages can be found in the default source for packages and their dependencies — Python Package Index (PyPI).

### 4.1 Installation

```
pip3 install inforion
```

### 4.2 Upgrade

If you want to upgrade the inforion, please use:

```
pip install inforion --upgrade
```

### 4.3 Show Version

If you want to see the version you have installed, please use:

```
pip show inforion
```



---

## Jupyter Notebook to Windows 10

---

The last prerequisite is to install **Jupyter Notebook** using the following command:

```
python -m pip install jupyter
```

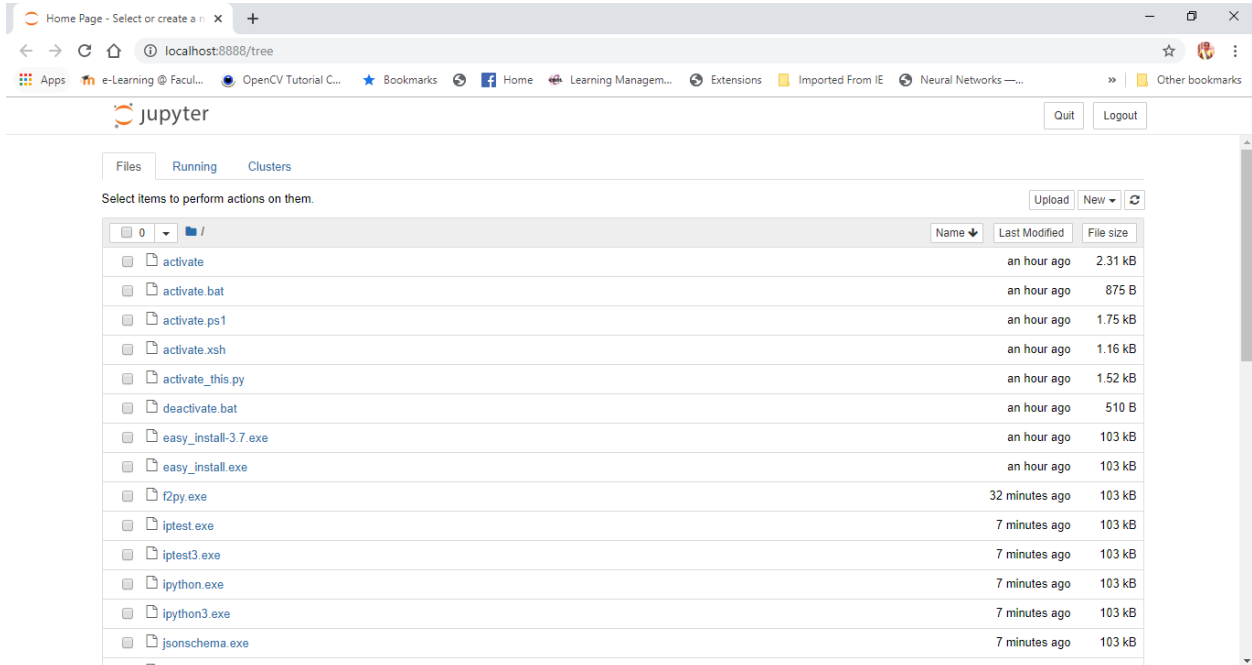




```
Command Prompt - jupyter notebook
C:\Users\Kalhari\opencv\Scripts>jupyter notebook
[I 06:41:07.787 NotebookApp] Writing notebook server cookie secret to C:\Users\Kalhari\AppData\Roaming\jupyter\runtime\notebook_cookie_secret
[I 06:41:08.913 NotebookApp] Serving notebooks from local directory: C:\Users\Kalhari\opencv\Scripts
[I 06:41:08.913 NotebookApp] The Jupyter Notebook is running at:
[I 06:41:08.914 NotebookApp] http://localhost:8888/?token=aebb5942cae32d7737b628619f97a189fd307b26bbe0f578
[I 06:41:08.915 NotebookApp] or http://127.0.0.1:8888/?token=aebb5942cae32d7737b628619f97a189fd307b26bbe0f578
[I 06:41:08.915 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 06:41:08.991 NotebookApp]

To access the notebook, open this file in a browser:
file:///C:/Users/Kalhari/AppData/Roaming/jupyter/runtime/nbserver-5276-open.html
Or copy and paste one of these URLs:
http://localhost:8888/?token=aebb5942cae32d7737b628619f97a189fd307b26bbe0f578
or http://127.0.0.1:8888/?token=aebb5942cae32d7737b628619f97a189fd307b26bbe0f578
```

Finally, start the notebook server and popup dashboard in browser using “localhost:8888/tree” url and now you will get access to the Jupyter Notebook.





---

## Generate M3 Excel Mapping File

---

### 6.1 Description

The **Excel mapping file generation** part generates an excel mapping file for a particular API program of M3.

It populates the mapping file with all the available fields for the specific M3 API program. This will be used by the Infor Consultant or the customer to map the M3 fields with the source database fields.

Once the mappings are specified, this mapping file is used to validate data from the source file (provided by the customer) and load it to M3.

*For more information and help, just enter this command to see the necessary parameters:*

```
inforion extract --help
```

### 6.2 Parameters

Parameter	Description
-p or -program	This parameter is used to provide the API program for which mapping file should be generated. e.g MMS301MI, CRS610MI, AAS320MI
-o or -outputfile	This parameter is used to provide the output file path where the generated mapping file should be saved.

### 6.3 Example

```
inforion extract -p CRS610MI -o CRS610MI-Mappings.xlsx
```

**Note:** this command should always be executed in terminal as this needs user permissions, if you do not use the DM\_ION Workflow manager.



In this section we are going to take a look at the following **Data Catalog (datacatalog) commands**:

## 7.1 create

### 7.1.1 Description:

Add or update object metadata in the Data Catalog. Schema type of JSON and DSV requires that the name, type, and schema are provided, and you can optionally include the properties. For object type ANY requires the name and type only, and the schema and properties should be omitted.

### 7.1.2 Parameters:

Parameter	Description
-ionfile or -i	Infor IONAPI credentials file.
-name or -n	Object name.
-schema_type or -t	Object schema type. Example DSV or ANY.
-schema or -s	Schema file (JSON).
-properties or -p	Additional schema properties file (JSON).

### 7.1.3 Example:

```
$ inforion catalog create --ionfile credentials/credentials.ionapi --name CSVSchema2 -  
↪-schema_type DSV --schema data/catalog_schema.json --properties data/catalog_  
↪properties.json
```

## 7.2 delete

### 7.2.1 Description:

Delete a schema object by name.

### 7.2.2 Parameters:

Parameter	Description
-ionfile or -i	Infor IONAPI credentials file.
-name or -n	Name of the schema object.

### 7.2.3 Example:

```
$ inforion catalog delete --ionfile credentials/credentials.ionapi --name CSVSchema2
```

Now, it is time to look up the *Data Lake (datalake) commands*:

## 8.1 list

### 8.1.1 Description:

List data object properties using a filter.

### 8.1.2 Parameters:

Parameter	Description
-ionfile or -i	Infor IONAPI credentials file.
-list_filter or -f	The restrictions to be applied on the returned records.
-sort or -s	Field name followed by colon followed by direction (asc or desc; default asc). Example: ,event_date:desc'.
-page or -p	The page number from which to start returning records. Starts from 1.
-records or -r	The number of records that will be returned. Starts from 0

### 8.1.3 Example:

```
$ inforion datalake list --ionfile credentials/credentials.ionapi --list_filter "dl_
↳document_name eq 'CSVSchema2'"
```

## 8.2 get

### 8.2.1 Description:

Retrieve payload based on id from datalake.

### 8.2.2 Parameters:

Parameter	Description
-ionfile or -i	Infor IONAPI credentials file.
-stream_id or -i	Object ID.

### 8.2.3 Example:

```
$ inforion datalake get --ionfile credentials/credentials.ionapi -id 1-7e476691-b17c-  
↪3e8d-8f0c-ea13222f56ef
```

## 8.3 upload

### 8.3.1 Description:

This command use the ION Messaging Service to send documents into ION.

### 8.3.2 Parameters:

Parameter	Description
-ionfile or -i	Infor IONAPI credentials file.
-schema or -s	Schema name.
-logical_id or -l	The Logical Id of the sending application (fromLogicalId parameter).
--file or -f	File to upload.

### 8.3.3 Example:

```
$ inforion datalake upload --ionfile credentials/credentials.ionapi --schema_  
↪CSVSchema2 --logical_id lid://infor.ims.mongooseims --file data/sample.csv
```

## 8.4 purge

### 8.4.1 Description:

Deletes Data Objects based on the given filter or a list of IDs. You cannot use both parameters together: ids and purge\_filter.



### 8.4.2 Parameters:

Parameter	Description
-ionfile or -i	Infor IONAPI credentials file.
-ids or -id	Object ids.
-purge_filter or -f	The restrictions to be applied to purge the records.

### 8.4.3 Example:

```
$ inforion datalake purge --ionfile credentials/credentials.ionapi --ids 1-dd6aa276-  
↳b34d-3905-b378-cdb5452ca17f,1-02d3ed52-5602-36ac-b3b1-fa670dbfeb72
```

```
$ inforion datalake purge --ionfile credentials/credentials.ionapi -f "dl_id eq '1-  
↳d358de11-4658-3c2d-a6ec-88c028f46315'"
```



## Export LN Data

## 9.1 Description

This section is supposed to help and guide the user to export data from LN to Excel files using **Infor ION API**. Before getting to know the necessary parameters for exporting data, it is highly important to mention that **Infor ION API** only supports the following services:

- SalesOrder
- Business\_Partner\_v3

*Just as a friendly reminder again, whenever you need help and advice just enter the following command and you will see which parameters are needed for the option of **\*\*exporting\*\***:*

```
inforion ln ExportData --help
```

## 9.2 Parameters

Parameter	Description
-u, -url	The full URL to the API is needed. Please note you need to enter the full url like <code>.../LN/c4ws/services/SalesOrder</code> [required]
-i, -ionfile	IONFile is needed to login in to Infor OS. Please go into ION and generate a IONFile. If not provided, a prompt will allow you to type the input text. [required]
-c, -company	Company for which you want to export data. e.g 121. [required]
-s, -service_name	Service name for which you want to export data. See above for currently supported service names. [required]
-o, -outputfile	File as Output File - Data will be exported to this file.

## 9.3 Example

```
inforion ln ExportData -s BusinessPartner_v3 -u https://Xi2016.gdeinfor2.com:7443/  
↪infor/LN/c4ws/services/ -i LN.ionapi -c 121 -o BusinessPartners.xlsx
```

Anyway, here is a very self-explanatory video of how to handle the step of extracting data.

<https://asciinema.org/a/347871.svg>

### 10.1 Description

This step provides the functionality of transforming the **external data source** into a **dataset for M3**, based on the mapping file.

### 10.2 Parameters

Parameter	Description
-a or -mapping-file	This parameter is used to provide the mapping file based on which transformation will be done.
-b or -mainsheet	This parameter is used to define the main sheet which will contain the mapping fields for transformation.
-i or -inputfile	This parameter is used to provide the input data.
-o or -outputfile	This parameter is used to write the transformed data into a file.

### 10.3 Example

```
inforion transform -a sample_mapping.xlsx -b Sheet1 -i sample_data.xlsx -o output.xlsx
```



### 11.1 Description

One of the last and most important steps is to finally load the data from your initial Excelsheet to the desired Infor Application; as for example M3.

*Again, for any doubts or help just enter this command and so you will see the existing parameters for the option of **\*\*loading\*\***:*

```
inforion load --help
```

## 11.2 Parameters

Parameter	Description
-u, -url	The full URL to the API is needed. Please note you need to enter the full url like .../M3/m3api-rest/v2/execute/CRS610MI [required]
-f, -ion-file	IONFile is needed to login in to Infor OS. Please go into ION and generate a IONFile. If not provided, a prompt will allow you to type the input text. [required]
-p, -program	What kind of program to use by the load [required]
-m, -method	Select the method as a list [required]
-i, -inputfile	File to load the data. Please use XLSX or CSV format. If not provided, the input text will just be printed [required]
-o, -outputfile	File as Output File - Data are saved here for the load
-s, -start	Dataload can be started by 0 or by a number
-e, -end	Dataload can be end
-z, -configfile	Use a Configfile instead of parameters

## 11.3 Example

```
inforion load -u https://mingle-ionapi.eu1.inforcloudsuite.com/Tendat_DEV/M3/m3api-
↪rest/v2/execute -f FellowKey.ionapi -p CRS610MI -m "Add,ChgBasicData,ChgOrderInfo,
↪ChgFinancial" -i excel/T-KundenNeu1.xlsx -o load_full_200.xlsx -s 0 -e 2
```



---

## Contributing guidelines

---

Contributions are always welcomed!

### 12.1 Big changes

Please open a new issue to discuss or propose a major change. Not only is it fun to discuss big ideas, but we might save each other's time too. Perhaps some of the work you're contemplating is already half-done in a development branch.

### 12.2 Code style

We use PEP8, `black` for code formatting and `isort` for import sorting. The settings for these programs are in `pyproject.toml` and `setup.cfg`. Pull requests should follow the style guide. One difference we use from „black“ style is that strings shown to the user are always in double quotes (") and strings for internal uses are in single quotes (').

### 12.3 Tests

New features should come with tests that confirm their correctness.

### 12.4 New Python dependencies

If you are proposing a change that will require a new Python dependency, we prefer dependencies that are already packaged by Debian or Red Hat. This makes life much easier for our downstream package maintainers.

Python dependencies must also be GPLv3 compatible.

## 12.5 New non-Python dependencies

InforION uses several external programs for its functionality. In general we prefer to avoid adding new external programs.

## 12.6 Style guide: Is it InforION or inforion?

The program/project is InforION and the name of the executable or library is inforion.

## 12.7 Known ports/packagers

InforION has been ported to many platforms already. If you are interesting in porting to a new platform,

In this section we present how logging works on Inforion CLI and API.

User can set the log level in the following way:

```
export LOG_LEVEL = 'see log levels bellow'
```

Current available log levels are:

- CRITICAL
- ERROR
- WARNING
- INFO
- DEBUG

By default, it is logging the only errors, which is kind of a default behavior. However, we can change it if you insist.



# KAPITEL 14

---

## Indices and tables

---

- [genindex](#)
- [modindex](#)
- [search](#)