
InforWorkflow

Release 2.1

Daniel Jordan

2020-07-16

USAGE

| | | |
|----------|--|-----------|
| 1 | Introduction | 3 |
| 2 | Install | 5 |
| 2.1 | Start on Mac or Linux | 6 |
| 2.2 | Start on Windows | 6 |
| 2.3 | Upgrade to Latest Build | 6 |
| 3 | How to run | 7 |
| 3.1 | Install Docker on Windows | 7 |
| 3.2 | How to compare results using EphesoftAutomated Docker instance | 9 |
| 4 | Requirements | 11 |
| 5 | Indices and tables | 13 |

Compares results extracted from Ephesoft with original/expected values to check the extraction accuracy.

INTRODUCTION

InforFlow is a wrapper library to create Prefect based flows to migrate data from MS-SQL Sever to Infor M3.

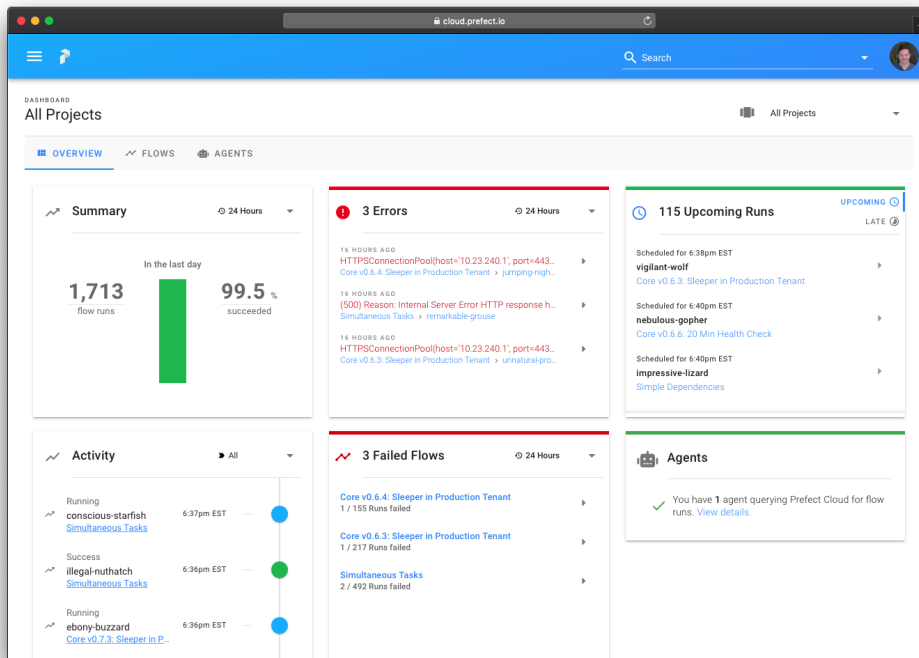
It is using Prefect library to create flows and uses InforION library to tranform and load data to M3.

INSTALL

To install the docker image please use

```
docker run fellowconsulting/inforion
```

Dockerhub Link



In our docker container we use Prefect for the UI to display the Workflows

For example start the docker container

Before you start check if you have created a conf directory. In the directory you should have following files:

- Mapping_Kundenstamm.xlsx
- credentials.yml
- IONAPI Key File
- module_dependencies.csv
- modules.csv

2.1 Start on Mac or Linux

```
docker run -v $PWD/conf:/ion-workflow/conf fellowconsulting/inforion flow -b /ion-  
↳workflow/conf -c credentials.yml -m modules.csv -d module_dependencies.csv
```

2.2 Start on Windows

```
docker run -v %cd%/conf:/ion-workflow/conf fellowconsulting/inforion flow -b /ion-  
↳workflow/conf -c credentials.yml -m modules.csv -d module_dependencies.csv
```

2.3 Upgrade to Latest Build

```
docker pull fellowconsulting/inforion:latest
```

HOW TO RUN

EphesoftAutomated is available as a Docker instance.

Docker is a platform for running containers with prepackaged applications. It's a system that we are going to use for packaging and deploying different versions and microservices.

3.1 Install Docker on Windows

Install Docker Desktop on Windows.

System Requirements:

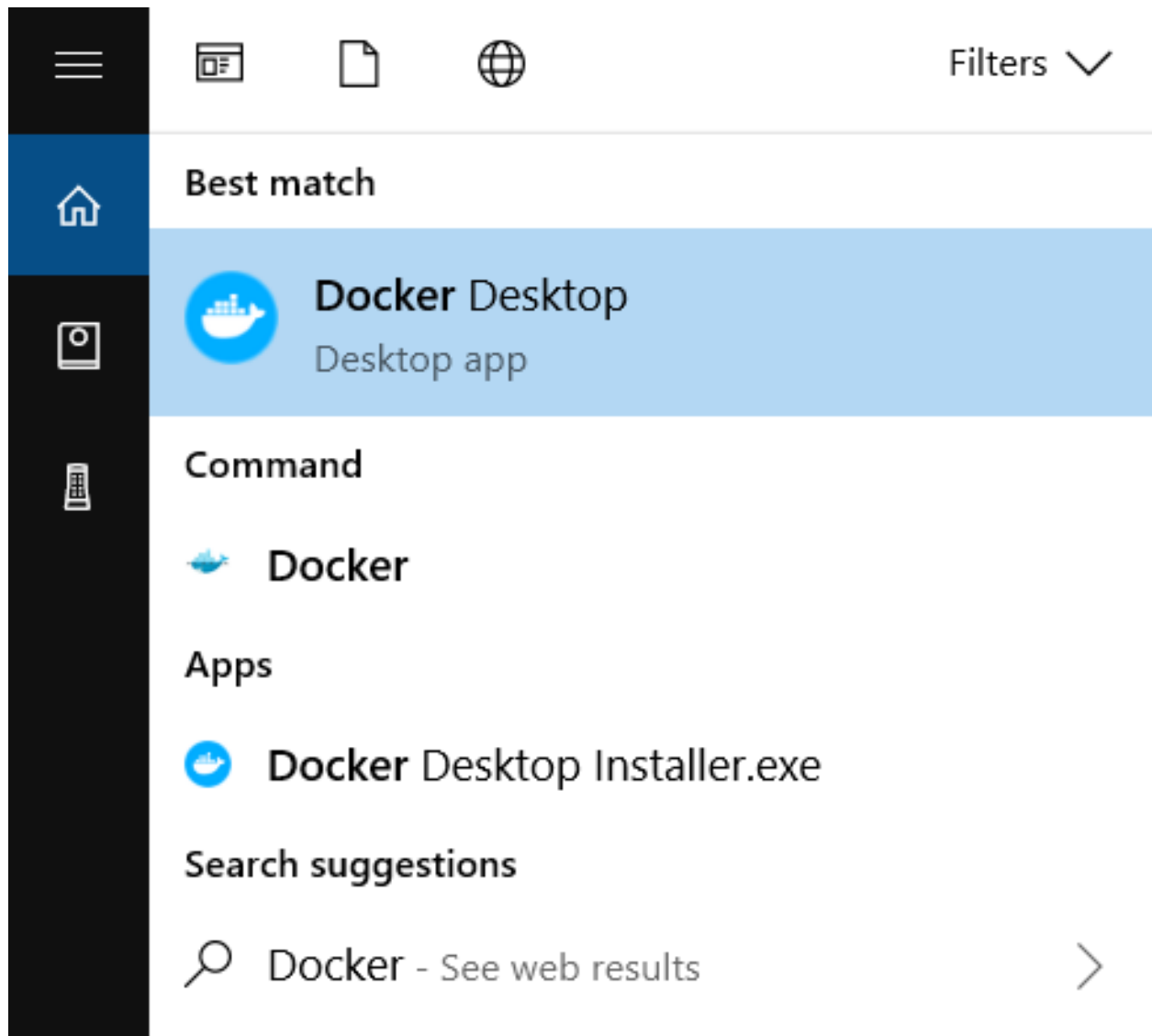
- Windows 10 64-bit: Pro, Enterprise, or Education (Build 16299 or later).
- Hyper-V and Containers Windows features must be enabled.
- The following hardware prerequisites are required to successfully run Client Hyper-V on Windows 10:
 - 64 bit processor with Second Level Address Translation (SLAT)
 - 4GB system RAM
 - BIOS-level hardware virtualization support must be enabled in the BIOS settings. For more information, see Virtualization.

Installation:

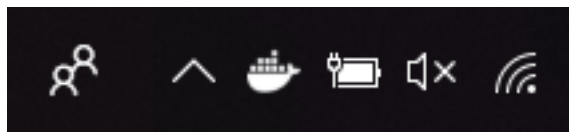
1. Double-click Docker Desktop Installer.exe to run the installer.
2. When prompted, ensure the Enable Hyper-V Windows Features option is selected on the Configuration page.
3. Follow the instructions on the installation wizard to authorize the installer and proceed with the install.
4. When the installation is successful, click Close to complete the process.

Start Docker Desktop:

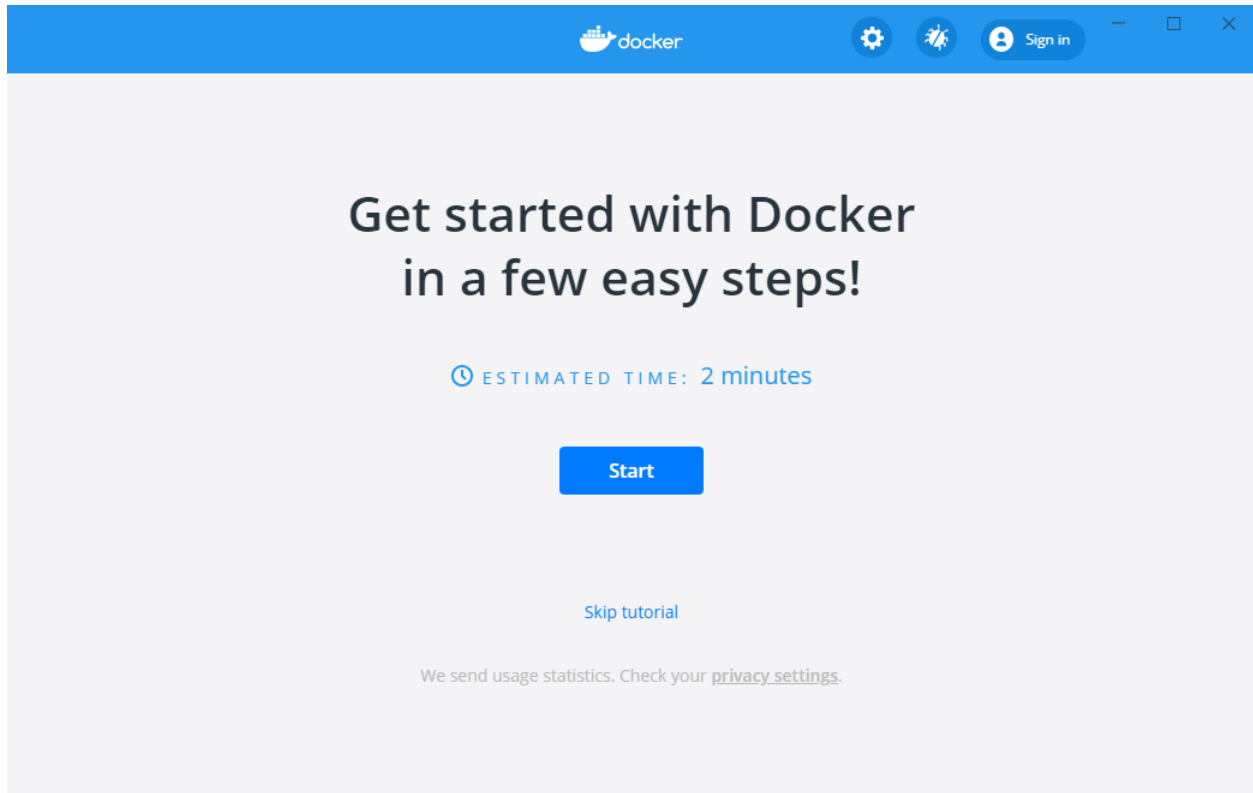
Docker Desktop does not start automatically after installation. To start Docker Desktop, search for Docker.



When the whale icon in the status bar stays steady, Docker Desktop is up-and-running.



Congrats! You are now successfully running Docker Desktop on Windows!



3.2 How to compare results using EphesoftAutomated Docker instance

```
docker run -v /local_path_to_conf_dir:/ephesoft-automation/conf ephesoftautomation_
↳compare -b /ephesoft-automation/conf -c config.yml
code . . .
```

- with **docker run** you can run your image as a container
- **-v** we are mounting our local directory to docker container, so it can use files from this directory.
- with **compare** we are adding the configuration files that are missing inside the Docker ecosystem

REQUIREMENTS

You need following information and documents before you can create flows

- Credentials
 - **Access to MS-SQL Server**
 - * Server name
 - * Username
 - * Password
 - * Source Database name (Database from which data will be transfered to M3)
 - * Staging Database name (Database where data will store after transformation)
 - **Access to Prefect cloud**
 - * Token for Prefect cloud
 - * Project name
 - **Access to M3**
 - * M3 URL
 - * ION API File

All these credential informations should be provided in **credentials.yml**

- Modules information

For every module you want to export data from MS-SQL Server to M3 you need following information and add this to a csv file. Ideally it should be named **modules.csv**.

 - Module name (You can put any name here. This is only used to create flows)
 - Program name (This should be valid M3 program name where data will be exported)
 - Mapping excel file path (Full path for the excel file which contains the sheet used for data tranformation)
 - Sheet name (Name of the sheet which contains all the transformation information)
 - Source table name: Table name from which data will be extracted.
 - Stage table name: Table name where data will stored after transformation.
- Modules dependencies information

You should know if a module you are exporting already depends on another module. e.g. OrderLineItems is dependent on Orders and should be transferred after Orders are transferred. This

dependencies need to be provided in a csv file in a parent-child format, ideally named **module_dependencies.csv**

INDICES AND TABLES

- `genindex`
- `modindex`
- `search`