
INTO-CPS Desktop Application

Release 4.0.0

Oct 21, 2020

Content

1	First Steps	3
1.1	Install the application	3
1.2	Installing toolchain dependencies	4
1.3	Performing a Co-Simulation	5
2	User Interface	7
2.1	Overview	7
2.2	Important Functionality	8
3	Integrations	9
3.1	Modelio	9
4	Tutorials	11
4.1	Simulating the Water Tank Example	11
5	Developer Documentation	13
5.1	Overview	14

Welcome to the INTO-CPS Desktop Application documentation. The best way to get started is to follow the [First Steps](#) guide which guides you through the installation process and the basics of the application.



For a guide on how to use the various features of the application [User Interface](#) provides an overview of how to access the most important functionality.

Note: The INTO-CPS Toolchain refers to a collection of programs not limited to the the desktop application. For an comprehensive overview of all tools take a look at the [INTO-CPS toolchain documentation](#)

The following sections guides you through the process of installing the application, installing its dependencies and finally demonstrates how a co-simulation can be configured and executed within the application.

1.1 Install the application

There are two options for installing the application. The recommended approach is to download the latest *Release Build*, which is may lack bleeding edge features, but is typically much more stable. The alternative, is building the latest *Development Build* from source, which gives access to the most recently added features, but at a potential cost of stability.

1.1.1 Release Build

The latest release of the application for Windows, Linux, and macOS can be found on [GitHub Releases](#). Download and extract the zip archive corresponding to your host architecture. The extracted directory should look similar to this:

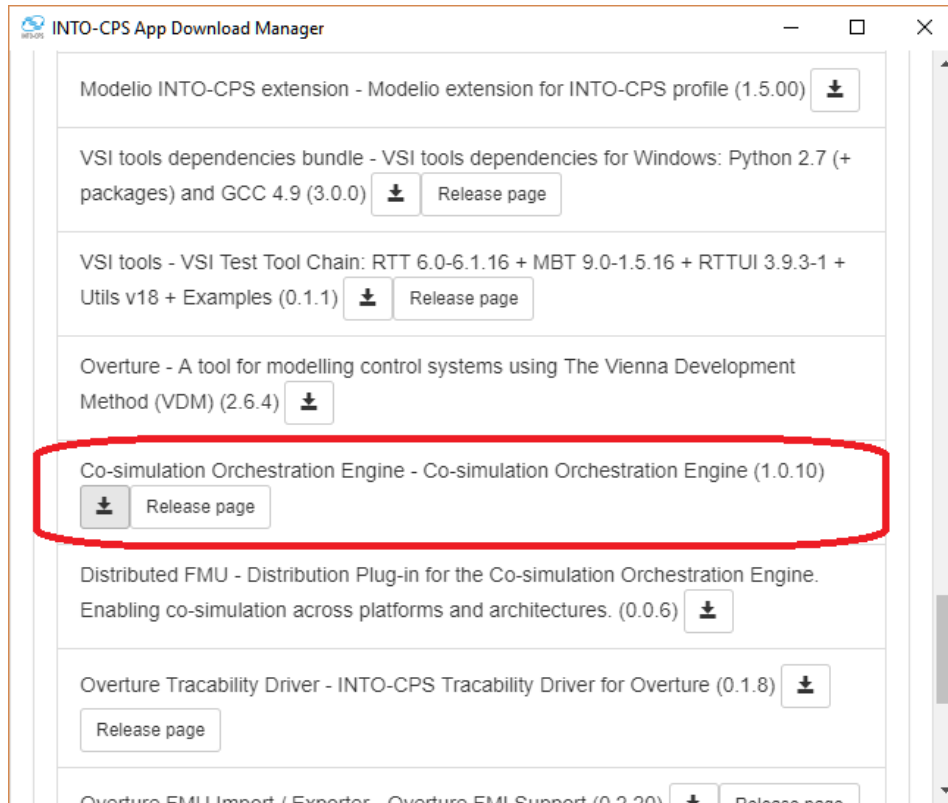
locales	4/17/2020 9:19 AM	File folder	
resources	4/17/2020 9:19 AM	File folder	
swiftshader	4/17/2020 9:19 AM	File folder	
chrome_100_percent.pak	12/20/2018 1:16 PM	PAK File	164 KB
chrome_200_percent.pak	12/20/2018 1:16 PM	PAK File	244 KB
d3dcompiler_47.dll	4/20/2018 7:29 AM	Application extens...	4,245 KB
ffmpeg.dll	12/20/2018 1:15 PM	Application extens...	2,077 KB
icudtl.dat	12/20/2018 1:03 PM	DAT File	9,979 KB
into-cps-app-4.0.0.exe	4/17/2020 9:19 AM	Application	91,589 KB
libEGL.dll	12/20/2018 1:14 PM	Application extens...	107 KB
libGLSv2.dll	12/20/2018 1:14 PM	Application extens...	4,984 KB
LICENSE	12/20/2018 12:43 ...	File	2 KB
LICENSES.chromium.html	12/20/2018 1:08 PM	Firefox HTML Doc...	1,948 KB
natives_blob.bin	12/20/2018 1:22 PM	BIN File	123 KB
osmesa.dll	12/20/2018 1:13 PM	Application extens...	2,881 KB
resources.pak	12/20/2018 1:16 PM	PAK File	8,517 KB
snapshot_blob.bin	12/20/2018 1:29 PM	BIN File	628 KB
v8_context_snapshot.bin	12/20/2018 1:29 PM	BIN File	1,017 KB
version	12/20/2018 12:43 ...	File	1 KB
VkICD_mock_icd.dll	12/20/2018 1:10 PM	Application extens...	339 KB
VkLayer_core_validation.dll	12/20/2018 1:14 PM	Application extens...	3,190 KB
VkLayer_object_tracker.dll	12/20/2018 1:13 PM	Application extens...	2,179 KB
VkLayer_parameter_validation.dll	12/20/2018 1:13 PM	Application extens...	2,790 KB
VkLayer_threading.dll	12/20/2018 1:14 PM	Application extens...	2,077 KB
VkLayer_unique_objects.dll	12/20/2018 1:13 PM	Application extens...	2,096 KB

1.1.2 Development Build

1. clone the repository: `git clone https://github.com/INTO-CPS-Association/into-cps-application/tree/development`
2. change working directory to repository: `cd into-cps-application`
3. install node dependencies: `npm install`
4. install other resources: `gulp init`
5. build ui: `gulp`
6. run the test: `npm test`

1.2 Installing toolchain dependencies

The application is modular by design, as such it relies on external programs to support its features. The easiest way to install these is using the built in download manager.



1. Start the application by executing the program `into-cps-app-x.x.x`
2. Press the “Window.Show Download Manager” tab, a window will appear
3. Select the latest version of the toolchain
4. Select and download “Co-simulation Orchestration Engine - Co-simulation”

Note: This guide only installs the dependency necessary to perform co-simulations inside the application. To access other features it may be necessary to install additional programs.

1.3 Performing a Co-Simulation

Warning: Missing! Here we should explain how to run, say the water tank example.

CHAPTER 2

User Interface

The INTO-CPS Application is the front-end of the INTO-CPS tool chain. The following screenshot shows how it looks.

Warning: Here we should explain in simple terms how to use the application. Important concepts such as multimodel-config and co-simulation configuration should be explained!

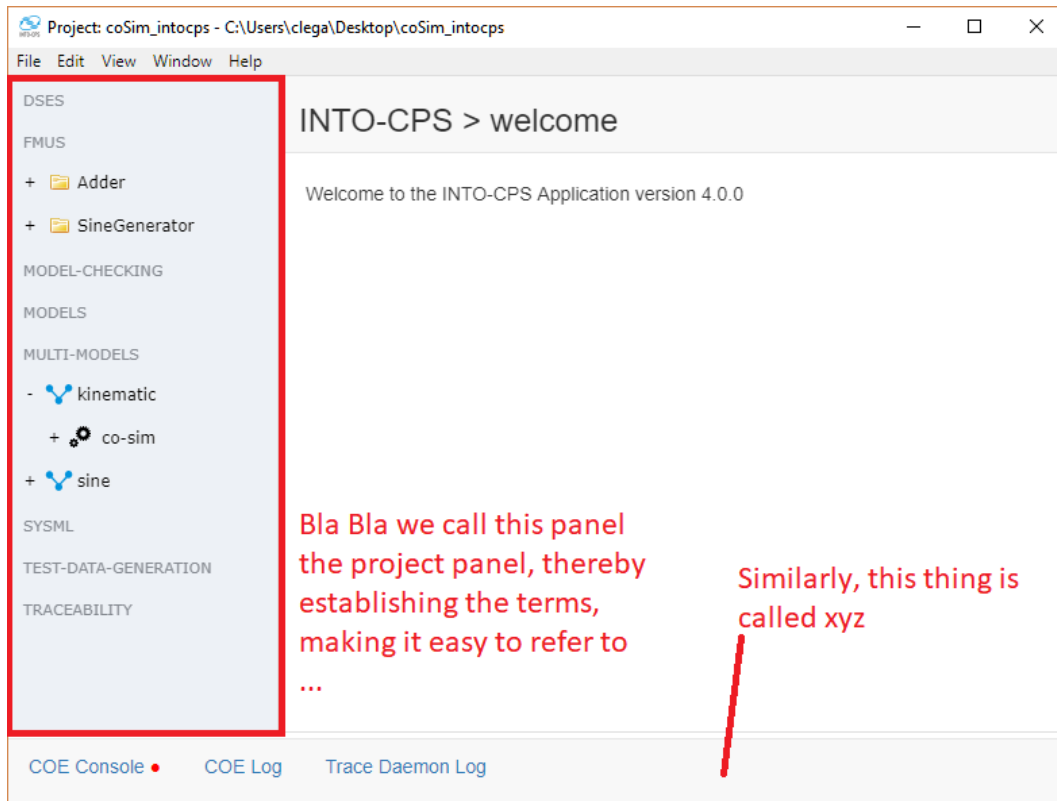
Additionally, a video demonstration would go a long way

Also we need need to consider what we are trying to convey. For example what information does the old docs convey:

The app **is** composed of 2 primary views: On the left, the project browser left **is** ↪ always visible. It shows the various components of an INTO-CPS project. These ↪ components can be clicked to trigger various actions. In the center, the main view **is** shown. The contents of this view depend on which ↪ action has been triggerred **in** the project browser. A few example views are shown ↪ below.

2.1 Overview

The app is composed of 2 primary views: On the left, the project browser left is always visible. It shows the various components of an INTO-CPS project. These components can be clicked to trigger various actions. In the center, the main view is shown. The contents of this view depend on which action has been triggerred in the project browser. A few example views are shown below.



2.2 Important Functionality

Warning: here we should describe how to use the various components of the application

2.2.1 Download Manager

2.2.2 Example Projects

2.2.3 Plotting

2.2.4 Design-Space Exploration

The application is built in a modular fashion that allows extra functionality to be accessed from within the application by installing external tools. Typically, these are installed using the *Download Manager*, however there are some cases where a plugin must be installed manually.

Below is an overview of the different integrations in terms of the functionality they provide and how to install them.

3.1 Modelio

Modelio is a combined UML/BPMN modeler supporting a wide range of models and diagrams.

Warning: integrate [this](#)

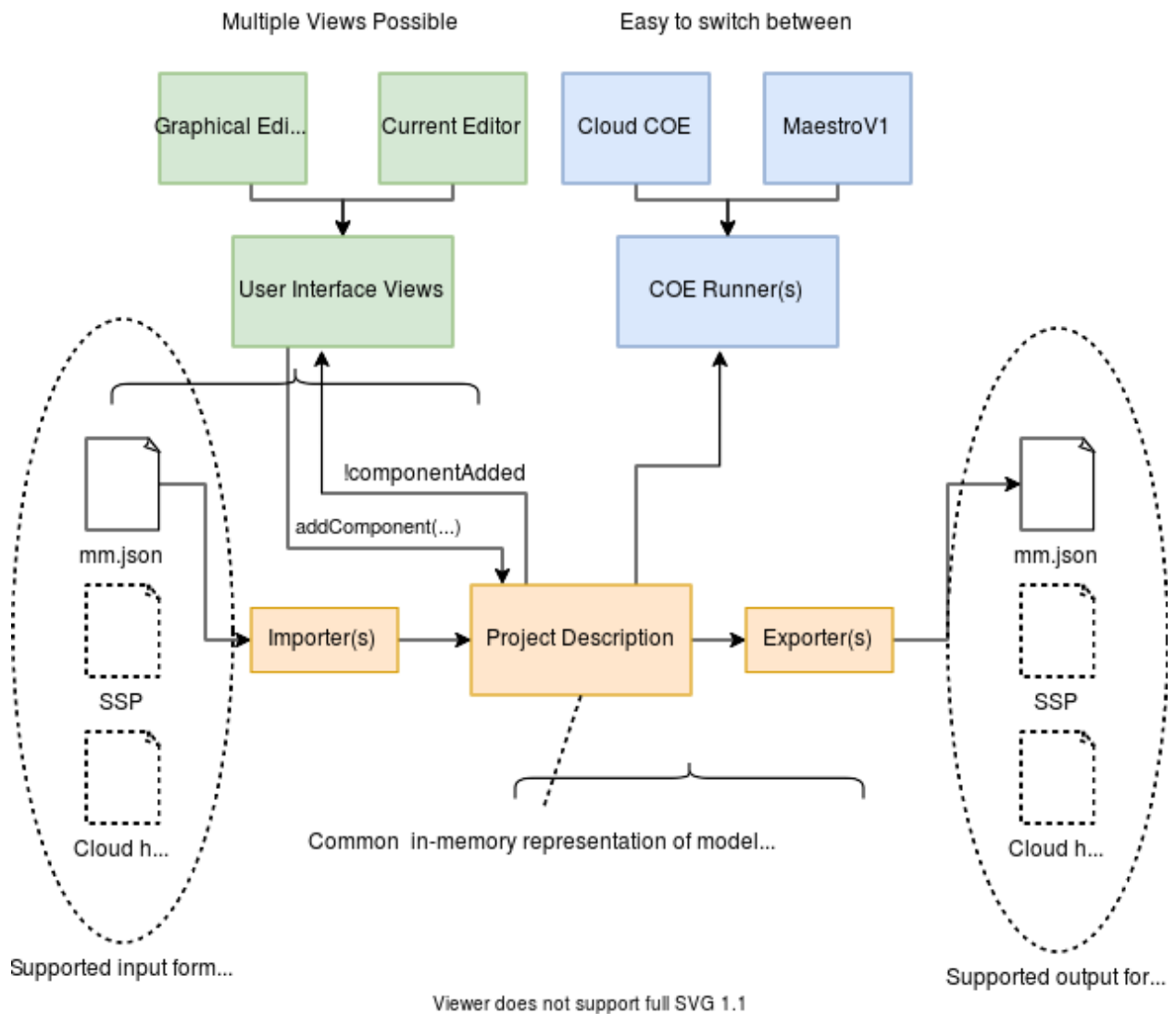
4.1 Simulating the Water Tank Example

CHAPTER 5

Developer Documentation

This section of the documentation provides relevant information to maintainers and contributors of the project.

5.1 Overview



test