IMPACT User's Guide
Version 0.1.0

IllinoisRocstar LLC

October 10, 2016

www.illinoisrocstar.com

License

The software package sources and executables referenced within are developed and supported by Illinois Rocstar LLC, located in Champaign, Illinois.The software and this document are licensed by the University of Illinois/NCSA Open Source License (see `opensource.org/licenses/NCSA`). The license is included below.

For more information regarding the software, its documentation, or support agreements, please contact Illinois Rocstar at:

- **tech@illinoisrocstar.com**
- **sales@illinoisrocstar.com**

0.1 Overview

IMPACT is a developing suite of software packages with an integrated build system. It is designed to be used as an infrastructure for composing multiphysics simulation capabilities from multiple, disparately developed simulation applications. The software is now fully functional and can be used to build multiphysics capabilities in its current state. The following quick start guide will help the user get started with obtaining, building, and using *IMPACT*.

0.2 How to Get *IMPACT*

IMPACT is distributed from its online repository, `https://github.com/IllinoisRocstar/IMPACT`. This distribution package includes the source and all available documentation. There are other files available for download which are not part of this integrated package.

This distribution directory structure for *IMPACT* contains:

```
IMPACT/
├── AUTHORS
├── LICENSE
├── README
├── CMakeLists.txt
├── Documentation/
│   ├── IMPACT_User.pdf
│   ├── COM_User.pdf
│   ├── SurfX_User.pdf
│   ├── SurfMap_User.pdf
│   ├── SIM_User.pdf
│   ├── SimIO_User.pdf
│   └── Simpal_User.pdf
├── COM/
├── SIM/
├── SimIO/
├── Simpal/
├── SurfMap/
├── SurfUtil/
└── SurfX/
```

This directory is the *source directory* of the distribution. It is highly recommended, and this document will assume, that the user will set an environment variable to indicate the *IMPACT*source directory:

```
MPACT_SOURCE=/the/full/path/to/IMPACT
```

The software packages and their documentation are included in the *IMPACT*distribution. In addition, the user will find the AUTHORS file which indicates the primary architects and developers for the suite of tools included in *IMPACT*. The LICENSE file contains the license text, which can also be found at `http://opensource.org/licenses/NCSA`. The README file is a very quick-and-dirty instruction on building *IMPACT*.

0.3 Build *IMPACT*

IMPACT uses Kitware's *CMake* build system, and should build on just about any Unix or Unix-clone system. We've tested it under the GCC, and Intel compilers on several flavors of Linux and MAC OS X. *IMPACT* requires *CMake*-2.8+.

0.4 Prerequisites and TPLs

IMPACT and its various packages have very few dependencies on outside packages. *IMPACT* requires the following support software and third-party libraries:

- C, C++, and Fortran 90 compilers. GCC and Intel are tested.
- *MPICH*-derived MPI can be obtained from `http://www.mpich.org` or `http://mvapich.cse.ohio-state.edu/` for MPI over InfiniBand. As a side note, the system should work with *OpenMPI* but is not well tested. MPI can also be installed by using a Linux system's software package manager.
- *HDF4* can be obtained from `http://www.hdfgroup.org/products/hdf4/` or installed by using Linux distribution package managers.
- *CGNS* (optional) can be obtained from `http://cgns.sourceforge.net/` or installed by using Linux distribution package managers.

These packages will need to be installed before building *IMPACT*.

0.5 Run *CMake*

With the `MPACT_SOURCE` directory all set up and prerequisites installed, the *IMPACT* software is ready to be configured and built. Out-of-source builds are highly recommended for *IMPACT*. To accomplish this, create a build directory that is not a subdirectory of `IMPACT_SOURCE`. It is assumed the user will create an environment variable to store the build directory:

```
IMPACT_BUILD=/full/path/to/mpact_build_directory
```

The user should set the following variables for specification of the compilers:

`CC=mpicc` (or equivalent)
`CXX=mpicxx` (or equivalent)
`FC=mpif90` (or equivalent)

If *HDF4* is installed somewhere other than a system location (e.g., /usr, or /usr/local), then an additional variable should be set:

```
CMAKE_PREFIX_PATH=/path/to/hdf4/install
```

Where *HDF4* libraries should be found in /path/to/hdf4/install/lib. Once these are set up, the user can go into the build directory and run *CMake*:

```
> cd ${IMPACT_BUILD}
> cmake ${IMPACT_SOURCE}
```

If all of the necessary prerequisites are satisfied, the stated commands should complete without error, resulting in *Makefile*s customized for the host environment. To build *IMPACT*, just issue "make":

```
> make
```

This command should complete the build without errors. Presuming this happened, the user will find the following libraries in the build tree:

```
lib/
├── libSITCOM.so
├── libSITCOMF.a
├── libRHDF4.so
├── libSimIN.so
├── libSimOUT.so
├── libSimpal.so
├── libSIM.so
├── libSurfMap.so
├── libSurfUtil.so
└── libSurfX.so
```

0.6 Use *IMPACT*

IMPACT is not an application, but a software development infrastructure. It is designed to be used from the build directory, and provides its capabilities in libraries that are linked by the user's applications. Assuming the user's software package is named *UserFoo*, the following general steps are taken to integrate *UserFoo*:

1. Prepare the application for integration.
   (a) Massage *UserFoo* architecture so that it consists of a library and a driver which links and drives the library.
   (b) Major part: Make necessary changes to represent interacting interface surfaces as stand-alone, self-descriptive surface mesh.
   (c) Major part: Make necessary changes to support externally supplied boundary solution on interface meshes. The source of the solution and any transformations may be neglected.
2. Implement the Component-side Client in *UserFoo*'s library.
   (a) Implement `UserFoo_load_module` and `UserFoo_unload_module` (see *COM* User's Guide).
   (b) Using *COM* API, create *UserFoo*'s ComponentInterface and register *UserFoo*-native data and functions as needed.
3. Implement a driver making sure it does the following:
   (a) Initializes MPI, if necessary
   (b) Initializes *COM*
   (c) Loads *UserFoo* with `COM_LOAD_STATIC_DYNAMIC(UserFoo,"userfoowindow_name")`

(d) Can access registered DataItems through the CI

(e) Can call *UserFoo* functions through the CI

The *COM* API is the most used part of *IMPACT* for preparing and integrating an existing application. This API is documented in the *COM* User Guide. Once ready to develop a driver, *SIM* may be optionally used to create multiphysics drivers, or the driver can be entirely designed by the user. Again, the *COM* API is used to access integrated applications' data and methods through the *COM* CI. A user's driver may load and use any of the service modules included in *IMPACT* in creating a multiphysics capability. Each package is described in its own documentation included in the *IMPACT* distribution in IMPACT_SOURCE/Documentation.