SANDIA REPORT

SAND2018-9008
Unlimited Release
Printed August 2018

Instructions for the Installation and Testing on a Windows System of the Sandia Automatic Report Generator

Mériadeg Perrinel, Philippe P. Pébaÿ, Robert L. Clay

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia National Laboratories

Telephone:	(865) 576-8401
Facsimile:	(865) 576-5728
E-Mail:	reports@adonis.osti.gov
Online ordering:	http://www.osti.gov/bridge

Telephone:	(800) 553-6847
Facsimile:	(703) 605-6900
E-Mail:	orders@ntis.fedworld.gov
Online ordering:	http://www.ntis.gov/help/ordermethods.asp?loc=7-4-0#online

SAND2018-9008
Unlimited Release
Printed August 2018

Instructions for the Installation and Testing on a Windows System of the Sandia Automatic Report Generator

Mériadeg Perrinel, Philippe P. Pébaÿ
NexGen Analytics
30 N. Gould St, Suite 5912
Sheridan, WY 82801, U.S.A.
{meriadeg.perrinel,philippe.pebay}@ng-analytics.com

Robert L. Clay
Sandia National Laboratories
P.O. Box 969
Livermore, CA 94551, U.S.A.
rlclay@sandia.gov

Abstract

This report is a sequel to [PC18], where we provided the detailed installation and testing instructions of Sandia's currently-being-developed Automatic Report Generator (ARG), for both Linux and macOS target platforms.

In the current report, we extend these instructions to the case of Windows systems.

Acknowledgments

The authors would like to thank our Sandia colleagues who have been involved in this project, in alphabetical order: Ernest "Foss" Friedman-Hill, Ed Hoffman, George Orient, and Nathan Spencer.

Contents

This page intentionally left blank

1 Generalities

The goal of this report is to provide detailed and reproducible installation instructions for the Automatic Report Generator (ARG), currently being developed at NexGen Analytics with funding and support from Sandia National Laboratories as part of the ASC Integrated Workflow Project (IWP), cf. [iwf18].

Specifically, this document extends [PC18], which focused on Linux and macOS targets, by reporting on the installation of the ARG on Windows systems.

This document contains several occurrences of the "`...`" string prepending the remainder of a path, for instance:

```
...\VTK-7.1.1\install\lib\python2.7\site-packages
```

in §1.2. This indicates that the user must replace the string "`...`" with the prefix path of the corresponding module on the system. For exemple, in the case mentioned above, `...` may be replaced with `C:\Dev\`; in that case, the user must read

```
C:\Dev\VTK-7.1.1\install\lib\python2.7\site-packages
```

instead of

```
...\VTK-7.1.1\install\lib\python2.7\site-packages
```

1.1 Obtaining the ARG

In order to obtain the ARG, you need to be granted access by one of the authors of this report: please contact either of them in this regard.

Once access has been granted, the ARG may be downloaded via the web interface of GitLab at `https://gitlab.com/iwf/arg`. Alternatively, you can install the Git BASH application for Window at `https://git-scm.com/download/win`, then launch it and issue the following command:

```
git clone git@gitlab.com:iwf/arg.git
```

Please note that when working from behind a firewall, the `https_proxy` variable must be set (ideally, added to the default user shell profile) in order to allow for the cloning command to go through said firewall.

As a result of either method, you will obtain (after unpacking the downloaded tarball in the former approach) a folder called `IWF`, directly under which a sub-folder called `ARG` is located.

Of particular interest in the context of this report are the various cases containted in the `tests` sub-directories.

1.2 Dependencies

The ARG is written in Python, version 2, and is known to work with versions 2.7.10 and above thereof. It is therefore required that the target platform provide such a version of Python as no pre-compiled binaries of the ARG are currently distributed.

In addition, the ARG currently only features a LaTeX backend (although alternatives will be offered in the future). It is therefore required that a reasonably complete installation of LaTeX be available on the target system. Please note that we have successfully tested the ARG on Window the MikTeX distributions, available at `https://miktex.org/download`.

Aside from these the following dependencies are either required or optional:

1. Required:
   (a) SEACAS, which can be (without prior authentication):
      - either downloaded at `https://github.com/gsjaardema/seacas`
      - or fetched interactively with Git BASH as follows:

        ```
        git clone https://github.com/gsjaardema/seacas.git
        ```

   (b) Exomerge, which can be (without prior authentication):
      - either downloaded at `https://github.com/gsjaardema/seacas`
      - or fetched interactively with Git BASH as follows:

        ```
        git clone https://github.com/timkostka/exomerge.git
        ```

   (c) The following Python packages:
      i. `numpy`
      ii. `pyyaml`
      iii. `pylatex`
2. Optional:
   (a) The Visualization Toolkit (VTK, cf. [VTK10]), if it is desired that the ARG produce 3-dimensional visualizations of data sets. In the context of the ARG, it is recommended to build VTK from source (a process that we explain in the subsequent chapters on a per-platform basis), using a recent stable version, but not too recent in order to avoid problems due to the requirement of using C++11 with versions 8 and above. We therefore settled with version 7.1.1, and the remainder of this report is written under this assumption. Note that after installing VTK, it is necessary to add `...\VTK-7.1.1\install\lib\python27\site-packages` to the existing path variable, using the Windows system variables dialog.

(b) Octave or MATLAB, in order to allow for the execution of N. Spencer's log file post-processing toolset. Please note that all work reported here has been done with the Octave; although the post-processing routines were originally written specifically for MATLAB, they have proven to be fully portable to the free software alternative. We have validated this approach with version 4.2.2 of Octave, obtained at `https://www.gnu.org/software/octave/#install`; in that case, `...\Octave-4.2.2\bin` must also be added to the environment path variable.

(c) The following Python packages:

   i. `matplotlib`, if it is desired that the 2-dimensional plotting capabilities of the ARG be made available,

   ii. `oct2py`, in order to allow for the Octave to Python bridging between the ARG and the aforementioned log file post-processing routines,

   iii. `scipy`, which at least with some distributions appears to be a dependency of `oct2py` not accounted for by Pip.

Regarding Python packages, we recommend you install those system-wide, using the Pip package installer utility, from a Windows terminal (`CMD`) with the following command issued under the `Scripts` subfolder in the Pip path:

```
pip2.7.exe install <package-name>
```

2 Windows Specifics

The deployment described here was made on a Windows version 10 system, with Visual Studio 15 (2017) 64-bit, and CMake version 3.10.0.

We recommend, for each library to be installed, that you create a build folder under the source one, in which the compilation will be performed. We also suggest to create an installation folder, whose aim is to contain the results of the `install` make target. For instance when using CMake, one can specify the install directory by specifying the `CMAKE_INSTALL_PREFIX` variable.

For each library, once configured, you will have to open the visual solution generated in the `...\src\build` directory. Visual Studio 2017 should recognize it and allow it to proceed with compilation and installation. Finally, the installation *per se* can be done by generating the `INSTALL` target that is loaded by Visual Studio and can be viewed in the solution explorer.

2.1 Environment Settings

We are working under the assumption that all system-wide installations performed to meet the general requirements of §1 were built and/or installed as shared libraries, within the the sub-folder tree to which the `PATH` environment variable is pointing, so that these libraries or executables can be found by the linker.

In addition, a new `PYTHONPATH` environment variable must be created in the same dialog where `PATH` was created, to so the locally-installed Python packages may be found (e.g., `...\VTK-7.1.1\install\lib\python2.7\site-packages`).

2.2 Python Dependencies

The easiest way to obtain the Pip package installer for Python, version 2, is to run the following command from a `CMD` terminal:

```
python.exe -m pip install --upgrade pip
```

It is also necessary to add `...\python27` to the `PATH`, in the system variables dialog.

In addition, after Exomerge has been obtained as indicated in §1.2, `...\exomerge` must be added to the `PYTHONPATH` discussed above.

2.3 Perl

Download the source (validated with `Perl-5.24.3.2404`) from:

https://www.activestate.com/activeperl/downloads

and then configure, compile, and install it. Add ...\Perl64\site\bin to the environment PATH variable.

2.4 ImageMagick

Download the source (validated with 7.0.8-4-Q16) from

https://www.imagemagick.org/script/download.php

and then configure, compile, and install it. Add ...\ImageMagick-7.0.8-Q16 to the environment PATH

2.5 SEACAS

A pre-requisite to building SEACAS from source (a requirement as indicated in §1.2) is to obtain the necessary third-party libraries. We have followed the following steps:

- ZLIB: Download the source (validated with zlib-1.2.11) from:

  https://zlib.net/

  and then configure, compile, and install it.
- HDF5: Download the source (validated with hdf5-1.10) from:

  https://support.hdfgroup.org/ftp/HDF5/releases/hdf5-1.10/hdf5-1.10.2/src/

  and then configure it with:

  - HDF5_ENABLE_Z_LIB_SUPPORT enabled
  - ZLIB_INCLUDE_DIR set to ...\Seacas\TPL\zlib\zlib-1.2.11\install\include
  - ZLIB_LIBRARY_DEBUG set to ...\Seacas\TPL\zlib\zlib-1.2.11\install\lib\zlibd.lib
  - BUILD_STATIC_EXECS enabled
  - HDF5_USE_18_API_DEFAULT enabled.

  Finally, compile and install it.
- CURL: Download the source (validated with curl-7.59.0) from:

`https://curl.haxx.se/download.html`

and then configure (disable `BUILD_TESTS`), compile, and install it.

- NetCDF: Download the source (validated with `netcdf-c 4.6.1`) from:

  `https://www.unidata.ucar.edu/downloads/netcdf/index.jsp`

  and then configure it with:

  - `USE_HDF5` enabled
  - `HDF5_DIR` set to `...\Seacas\TPL\hdf5\hdf5-10\install\cmake/hdf5`
  - `HAVE_HDF5` set to `...\Seacas\TPL\hdf5\hdf5-10\install\include`
  - `CURL_LIBRARY` set to `...\Seacas\TPL\curl\curl-7.59.0\install\lib\libcurl-d_imp.lib`
  - `CURL_INCLUDE_DIR` set to `...\Seacas\TPL\curl\curl-7.59.0\install\include`
  - `BUILD_TESTING` disabled
  - `ENABLE_TESTS` disabled.

  Finally, compile and install it.

Once the third-party libraries are installed, you may start the process of building and installing SEACAS, as follows:

- Currently, SEACAS needs to be patched in order to compile on Windows:

  1. open `...\Seacas\packages\seacas\libraries\exodus\include\exodusII.h` in your favorite editor, and replace line 421:

     ```
     #define EXODUS_EXPORT extern
     ```

     with:

     ```
     #define EXODUS_EXPORT __declspec(dllexport)
     ```

     and then save and close this file.
  2. open `...\Seacas\packages\seacas\scripts\exodus.py.in`, paying attention to the Python-ic indentation (two spaces), replace:

     ```
     if os.uname()[0] == 'Darwin':
       EXODUS_SO = "@ACCESSDIR@/lib/libexodus.dylib"
     else:
       EXODUS_SO = "@ACCESSDIR@/lib/libexodus.so"
     EXODUS_LIB = cdll.LoadLibrary(EXODUS_SO)
     ```

     with:

```
if os.uname()[0] == 'Darwin':
  EXODUS_SO = "@ACCESSDIR@/lib/libexodus.dylib"
elif os.name == 'nt':
  EXODUS_SO = "@ACCESSDIR@/bin/exodus.dll"
else:
  EXODUS_SO = "@ACCESSDIR@/lib/libexodus.so"
EXODUS_LIB = cdll.LoadLibrary(EXODUS_SO)
```

and then save and close this file.

- The configuration step is then done, using cmake-gui, as follows:

```
BUILD_SHARED_LIBS=ON
CMAKE_BUILD_TYPE=DEBUG
CMAKE_CONFIGURATION_TYPES=Debug
SEACAS_ENABLE_SEACAS
SUPES=OFF
SEACASProj_ENABLE_SEACASExodus=ON
SEACASProj_ENABLE_SEACASExodus_for=OFF
Netcds_LIBRARY_DIRS = .../netcdf/install/lib
Netcds_INCLUDE_DIRS = .../netcdf/install/include
```

- Compile and install the library, and add `...\Seacas\install\bin` to the environment `PATH` variable.
- Finally, copy all dynamic link libraries (`*.dll` files) of the aforementioned SEACAS dependencies (Exodus, CDF, Exodus, HDF5, CURL, ZLIB) into the `...\Seacas\Install\bin` folder.

Remark: When configuring Exodus, the `ExoIIv2for32` and `Ioss` modules suffice to generate a partial build of SEACAS that satisfies the ARG dependencies. This may be verified by inspecting the CMmake configuration log of SEACAS, that should display the following:

```
Processing enabled package: SEACAS
Processing enabled package: SEACAS (Exodus, ExoIIv2for32, Ioss)
```

Please note that the compilation will not complete if you attempt to compile the entire SEACAS projet, but its Exodus part will and that is all that is needed by ARG.

2.6 LaTeX

The simplest way to go about fully installing MikTeX on your target Windows platform is by downloading it from `https://miktex.org/download` (validated with version 2.9).

You will then need to add `...\MiKTeX 2.9\miktex\bin\x64\` to the environment `PATH`.

2.7 VTK

VTK itself can be built relatively easily, as fully explained in the download and build instructions at `https://www.vtk.org/Wiki/VTK/Configure_and_Build`, and must also be installed system-wide. Because all general instructions can already be found in the aforementioned online instructions, we only mention here what is specific to the ARG context; specifically, here are the CMake variables that must be adjusted, using cmake-gui:

```
VTK_ENABLE_VTKPYTHON=ON
VTK_WRAP_PYTHON=ON
VTK_PYTHON_VERSION=2.7
PYTHON_EXECUTABLE=.../Python27/python.exe
PYTHON_INCLUDE_DIR=.../Python27/include
PYTHON_LIBRARY=.../Python27/libs/python27.lib
```

Once the VTK configuration and `Makefile` generation process have been performed with CMake, compile and install it. Finally, add `...\VTK-7.1.1\install\bin` to the `PATH`, in the system variables dialog.

3 Testing and Support

In order to test your deployment of the ARG, you may either individually exercise any of the provided test cases, or launch the entire test harness at once.

In the former case, for instance in order to generate the `RanVib` case, launch a `CMD` terminal and move to the corresponding folder:

```
cd ...\IWF\ARG\tests\RanVib
```

which intially contains only some examplar data, together with a minimal `constants.in` file which you may also modify in order to experiment with the test case, or directly execute the following:

```
python.exe ...\IWF\ARG\src\Explorator.py -d data -s test
```

in order for the Explorator component of the ARG to generate a file called `test.yaml`, which describes the report document structure in abstract form based on the discovered data. Subsequently, execute:

```
python.exe ...\IWF\ARG\src\ReportAssembler.py -s test.yaml -d data
```

so the ReportAssembler component of the ARG may create the actual report in both LaTeX and PDF formats, along with all necessary artifacts.

On the other hand, if you want to launch the entire test harness at once, also starting from a `CMD` terminal:

```
cd ...\IWF\ARG\tests
make clean
make
```

You will then see several dozens of lines of text output, detailing exactly the operations that the ARG, in its various components, is performing.

As a final result, you shall obtain the following six SAND reports in PDF format:

```
SAND2018-DropPackage.pdf
SAND2018-ModalPackage.pdf
SAND2018-ModalPackage_2.pdf
SAND2018-PLOTS.pdf
SAND2018-can.pdf
SAND2018-disk.pdf
```

In case you encounter any difficulty in this process, or the SAND reports are not or incorrectly generated, please provide detailed feedback to the authors of this document.

References

[iwf18] IWF Confluence Wiki, 2018. https://snl-wiki.sandia.gov/display/IWF.

[PC18] P. Pébaÿ and R. Clay. Installation and testing instructions for the sandia automatic report generator (ARG). Sandia Report SAND2018-4421, April 2018. http://prod.sandia.gov/techlib/access-control.cgi/2018/184421.pdf.

[VTK10] *The VTK User's Guide, version 5.4*. Kitware, Inc., 2010.

DISTRIBUTION:

2	MS 9018	Central Technical Files, 8944
1	MS 0899	Technical Library, 9536

v1.40

Sandia National Laboratories