



PNT Integrity Toolkit Documentation

v3.2.1

Generated by Doxygen: September 23, 2021

# Contents

<b>1</b>	<b>Demonstration Toolkit for the PNT Integrity Library</b>	<b>3</b>
<b>2</b>	<b>Licenses</b>	<b>15</b>
<b>3</b>	<b>Build Instructions for the PNT Integrity Toolkit</b>	<b>27</b>
<b>4</b>	<b>Namespace Index</b>	<b>31</b>
4.1	Namespace List . . . . .	31
<b>5</b>	<b>Class Index</b>	<b>33</b>
5.1	Class List . . . . .	33
<b>6</b>	<b>File Index</b>	<b>35</b>
6.1	File List . . . . .	35
<b>7</b>	<b>Namespace Documentation</b>	<b>37</b>
7.1	csv_write Namespace Reference . . . . .	37
7.1.1	Detailed Description . . . . .	38
7.1.2	Function Documentation . . . . .	38
7.1.2.1	toString() [1/5] . . . . .	38
7.1.2.2	toString() [2/5] . . . . .	38
7.1.2.3	toString() [3/5] . . . . .	38
7.1.2.4	toString() [4/5] . . . . .	39
7.1.2.5	toString() [5/5] . . . . .	39
7.2	integrity_toolkit Namespace Reference . . . . .	39
7.2.1	Detailed Description . . . . .	40

<b>8</b>	<b>Class Documentation</b>	<b>41</b>
8.1	csv_write::MsgCsvWrite Class Reference	41
8.1.1	Detailed Description	42
8.1.2	Member Function Documentation	42
8.1.2.1	closeFile()	42
8.1.2.2	isFileOpen()	42
8.1.2.3	openFile()	42
8.1.2.4	setTimestampsEnabled()	43
8.1.2.5	setVerboseOutput()	43
8.1.2.6	writeHeader()	44
8.1.2.7	writeLine() [1/2]	44
8.1.2.8	writeLine() [2/2]	44
8.2	integrity_toolkit::ToolkitApplication Class Reference	45
8.2.1	Detailed Description	45
8.3	integrity_toolkit::ToolkitLCDDisplay Class Reference	45
8.3.1	Detailed Description	46
8.3.2	Constructor & Destructor Documentation	46
8.3.2.1	ToolkitLCDDisplay()	46
8.3.3	Member Function Documentation	47
8.3.3.1	displayCheckState()	47
8.3.3.2	initialize()	47
8.3.3.3	SetAssuranceLevel()	47
<b>9</b>	<b>File Documentation</b>	<b>49</b>
9.1	include/integrity_toolkit/CsvWrite.hpp File Reference	49
9.1.1	Detailed Description	50
9.2	include/integrity_toolkit/ToolkitApplication.hpp File Reference	50
9.2.1	Detailed Description	52
9.3	include/integrity_toolkit/ToolkitLCDDisplay.hpp File Reference	52
9.3.1	Detailed Description	52
9.4	src/toolkitMainApplication.cpp File Reference	53
9.4.1	Detailed Description	53
	<b>Index</b>	<b>55</b>

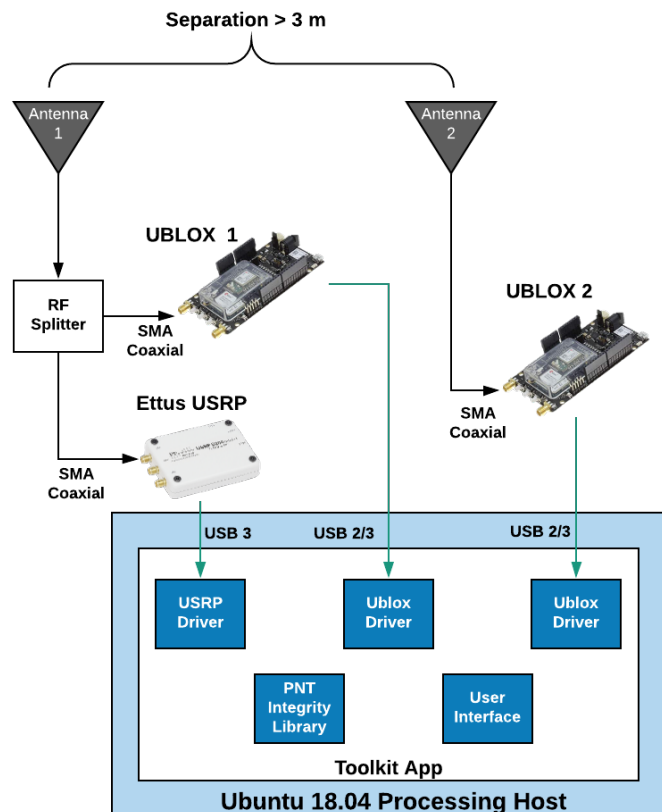


## Chapter 1

# Demonstration Toolkit for the PNT Integrity Library

The following document describes how a perspective end-user of the PNT Integrity Library can assemble a demonstrational toolkit with commercial-off-the-shelf (COTS) hardware. IS4S has developed a standalone toolkit application that interfaces with the hardware to create a demonstrational platform that allows the user to exercise all the functionality of the library.

The toolkit requires two separate antenna feeds for the multi-antenna checks featured in the library. The primary antenna feeds both a primary receiver (Ublox F9) and an Ettus USRP SDR. The second antenna feeds the second receiver. Both receivers and the SDR are connected to a processing host running the Ubuntu 18.04 linux distributions. The processing host runs the toolkit application that interfaces all of the sensor drivers, the integrity library, and a user interface. The figure below shows a high-level diagram of the toolkit components and how they are connected.



## Required Hardware Components

To use the toolkit, the user must acquire the following hardware:

### Ublox F9 Development Kit (Qty 2)

A Ublox F9P receiver is required for the primary receiver (Ublox 1) in the toolkit. The F9P provides access to all of the needed observables to stimulate all of the single-antenna checks in the toolkit, with the exception of the Acquisition Check which is fed by the USRP. A second Ublox receiver is required for the multi-antenna checks. For the secondary receiver, a Ublox M8-T or later may be used. IS4S recommends using the [Ublox F9P evaluation kit](#) for both receiver roles in the toolkit. Eval kits can be acquired via [Digi-Key](#).

### Ettus USRP

An Ettus Research USRP SDR device is required to run the Acquisition Check in the toolkit. IS4S recommends using the [B200 mini](#) model for this application. Any Ettus USRP that is capable of providing I/Q samples of the GPS L1-CA frequency and code will work, however, the toolkit is pre-configured for the B200 mini. The B200 mini has a footprint approximately the size of a credit card and has a single data/power connection via USB 3 from the processing host.

### Processing Host

A processing host capable of running the Ubuntu 18.04 LTS linux distribution the toolkit software and with enough I/O to connect to the 2 receivers and the SDR. Refer to the build instructions for all system requirements and dependencies. After obtaining an appropriate host system, the user should follow the instructions for Ubuntu 18.04 and select the option to build the full kit when appropriate.

## Device Configuration

The following sections give a brief discussion on configuring the devices for use with the toolkit. The user should refer to the user manuals and guides for their respective devices if more information is needed.

### Ublox F9-P

No specific configuration for the Ublox receiver is required, except for the proper cabling of the receiver to the processing host. A standard USB-A to USB-micro cable should be used to connect the Ublox F9-P evaluation board to the host. The user should verify that each receiver is recognized as a serial port device after connecting.

```
is4s@integrity-nuc1:~$ ls /dev/ttyACM*  
/dev/ttyACM0 /dev/ttyACM2
```

The user should determine which listed port represents which receiver, and not that this can change when the host is restarted or when devices are connected in different orders. Although not required, it is recommended to use udev rules to establish consistency for the connected receivers.

## Ettus USRP

As mentioned previously, IS4S recommends the bus-series USRP devices from ettus. Refer to the "Getting Started" [guide](#) for setup of the USRP. If a different USRP device is used, please locate the appropriate setup guide for the device on the Ettus [website](#). Once the USRP has been properly connected and configured, ensure that the device is properly connected to the splitter with an appropriate 50-ohm RF coaxial cable (SMA-male on the end connected to the USRP). Note that prior to setup, the UHD API must be properly installed on the host system.

To verify proper operation of the USRP, use the following command to list all connected devices:

```
is4s@integrity-nucl1:~$ uhd_find_devices
```

This will yield the following output for a default configured B200-mini.

```
[INFO] [UHD] linux; GNU C++ version 7.4.0; Boost_106501; UHD_3.14.1.1-release
-----
-- UHD Device 0
-----
Device Address:
  serial: 31CB5DF
  name: B200mini
  product: B200mini
  type: b200
```

Note the name of the device, as it will be used to configure the driver in the toolkit application.

## Configuring the input files

Each device in the kit is configured via a yaml file. The following sections give some guidance to properly configure these files for proper toolkit operation.

### Ublox Receivers

The ublox configuration files ([ublox1.yaml](#) and [ublox2.yaml](#)) should not require any modification except entering the correct serial port for the corresponding receiver. The Ublox connection is a serial over USB connection, thus a port will enumerate when the device is connected (i.e. /dev/ttyACM0). Enter the appropriate string for each device in its respective configuration file.

```
port: /dev/ttyACM0
```

## USRP

The contents of the `usrp.yaml` file are shown below. The file is pre-configured for operation with the recommended B200 mini USRP, with the antenna feed after the splitter connected to the RX2 port of the USRP. If the USRP's device name is changed via any of the Ettus-provided UHD tools, the new name must be used. Also, if a networked or X-series device is used, the "device\_args" argument will be a formatted string that contains the IP address of the USRP. Likewise, the "subdev" argument that represents the daughterboard in the USRP must be formatted for the particular daughterboard in use. Refer to the USRP / UHD guides on the Ettus website for more information on correctly setting these arguments.

```
device_args: name=B200mini
subdev: A:A
```

The user must set the proper sample type and wire format for the USRP in use. If possible, IS4S recommends using 8-bit samples, which is only possible for certain USRP models.

```
# 0 - sc8, 1 - sc16
if_sample_type: 0
wirefmt: sc8
```

The antenna port should be set as either 'TX/RX' OR 'RX2', depending on what port has been connected on the device. 'RX2' is recommended.

```
antenna: RX2
```

In certain cases, the USRP may be connected to an external clock source, such as a 10 MHz standard in a laboratory environment. In this case, 'external' should be used. If the device is in standalone mode, use 'internal'.

```
clock_ref: internal
```

The 'center\_frequency' should be set to GPS-L1, 1.57542 GHz.

```
center_frequency: 1575420000
```

The recommended sampling frequency is 5 MS/s

```
sampling_frequency: 5000000
```

Maximum gain (different for each USRP model) is recommended, but can be adjusted if needed in the quantization calibration step. Maximum gain for the B200 is 75 dB.

```
gain: 75
```

The following parameters should not be modified unless you are encountering data throughput issues such as repeated overflow warnings.

```
samples_per_message: 10000
circ_buff_capacity: 100000
```

The time argument should be left as is to instruct the data streaming to run indefinitely.

```
total_time: 0.0
```



## Integrity

The `integrity.yaml` file contains all of the configuration parameters for the PNT Integrity Library that runs as part of the toolkit. The toolkit will run out of the box with no user modifications necessary. However, some of the parameters in the configuration file will most likely have to be properly tuned for optimum performance. These are discussed in more detail in the following sections. For detailed information on each check, refer to the PNT Integrity Library documentation.

### Common Parameters

All checks have an enable flag and a weighting parameter. If certain checks are not desired, they can simply be disabled by setting the flag to false, i.e.

```
enable_rng_pos_check: [true / false]
```

Similarly, a check's wait can be adjusted by

```
static_pos_check_weight: [value]
```

Additionally, each check has a level period in which once the level is elevated it cannot be lowered until that time period has expired. This is effectively a crude low-pass filter that prevents level "flickering" when key check metrics are at or near level change thresholds. The period can be set by

```
aoa_check_level_period: [value]
```

Each check also has a string name that is used for display and diagnostic purposes

```
acq_check_name: [string name for check]
```

### Angle of Arrival (AOA) Check

The single diff threshold and prn count thresholds are used to determine if multiple PRNs have a common angle of arrival. Each pseudorange value is differenced from all the others, and if there are a certain number of the differences are within a threshold, then that PRN is flagged as suspect. For example, the following settings can be interpreted as: "If there are at least 7 single differences for this PRN that are within 2 m of each other, then flag this PRN as suspect".

```
aoa_check_single_diff_thresh: 2.0  
aoa_check_prn_count_thresh: 7
```

After all PRNs have been examined, the check will add up the number of suspect PRNs and then threshold for the overall check level. The following settings can be interpreted as: "If there are at least 3 suspect PRNs then raise the level to "Inconsistent." If there are at least 5 then raise the level to "Unassured".

```
aoa_check_inconsistent_thresh: 3  
aoa_check_unassured_thresh: 5
```

For multi-node checks, the AOA check should not be run in the event that the nodes are close together. Therefore, this threshold will be used to disable the check if the reported inter-node range is below a threshold. For static applications, a constant value is used in place of a measured range between two nodes.

```
aoa_check_range_thresh: 3.0
```

Future versions of the AOA check will have the ability to optionally use carrier phase measurements in place of pseudo-ranges. For now, this argument must be set to "false."

```
aoa_check_use_carrier_phase: false
```

## Range-position Check

The range position check compares the reported distance between the two nodes (antennas) and the differenced receiver positions. For static applications, the baseline between the antennas should be known and entered into the settings appropriately.

```
enable_static_mode: true
rx1_rx2_baseline: 15.0
```

## Acquisition Check

The sampling rate for the acquisition check should match the sampling rate set for the USRP.

```
if_data_sampling_rate: 5000000
```

For all toolkit applications using an ETTUS USRP, the intermediate frequency should be set to 0.

```
if_data_intermediate_freq: 0.0
```

The following acquisition parameters should not be modified for the kit application:

```
acq_check_search_band: 10000.0
acq_check_search_step_size: 500.0
acq_check_integration_period: 0.001
acq_check_code_freq_basis: 1023000.0
acq_check_code_length: 1023
```

Assurance level thresholds for the acquisition check can be set with the following parameters. Similarly to the AOA check, the Acquisition check examines each PRN individually and then counts up the number of suspect PRRs to create an overall level for the check.

```
acq_check_inconsistent_thresh: 2.0
acq_check_unassured_thresh: 4.0
```

The following parameters must be specifically tuned by the end-user. A section devoted to this topic is presented later in this document.

```
high_power_threshold: 10000000
peak_ratio_threshold: 3.0
acquisition_threshold: 1000000
```

### Static Position Check

The static position, as its namesake implies, is only valid for static applications. The check simply monitors the position against a known, surveyed point. The check has the ability to perform the initial survey itself or it can be provided by the user.

For surveying, set the following flag accordingly:

```
survey_init: [true/false]
```

The user can also control the amount of samples taken before the surveyed position is computed. Update rate for position should be taken into account (typically 1 Hz)

```
num_pos_for_init: 60
```

If the user is providing a known surveyed position, set the "survey\_init" flag to false and provide the following information:

```
static_pos_lat_deg: [latitude in degrees, +/- for North/South]
static_pos_lon_deg: [longitude in degrees, +/- for Est/West]
static_pos_alt_m: [altitude in meters]
```

A moving time history window of positions is stored, and each position is differenced with the surveyed position and then thresholded with a desired value.

```
static_pos_check_window_size: 10
pos_change_thresh: 15.0
```

If a particular position difference is above the threshold, then it is flagged as suspect. A percentage of suspect points over the time history window is calculated and then thresholded with the following values to set the overall level for the check.

```
static_pos_check_inconsistent_thresh: 0.3
static_pos_check_unassured_thresh: 0.7
```

### Carrier-to-Noise Ratio (Cn0) Check

The Cn0 check examines the "flatness" of the Cn0 values of all PRNs over a given window. These parameters generally should not be changed.

### Position Jump Check

The position jump check examines the position for sudden jumps that are outside the bounds of the platforms dynamics. For the static toolkit application, the maximum velocity should be set to 0 indicating that there is no motion involved.

```
minimum_bound: 15
maximum_velocity: 0.0
```

## Position-Velocity Consistency Check

The settings for the PVC check are pre-configured for a Ublox and should generally not be changed.

## AGC Check

The AGC check values, similar to the Acquisition Check values, should be tuned by the user. This process is discussed in a later section

```
min_agc_value: 0.0
max_agc_value: 10000
agc_check_inconsistent_thresh: 0.4
agc_check_unassured_thresh: 0.5
```

## Clock Bias (Jump) Check

The settings for the clock bias check are pre-configured and should not be changed.

## Launching the Application

After [building](#) the application, the application is launched from the build folder

```
cd build
```

Providing the '-h' argument to the application will display the usage instructions

```
./integrity_toolkit -h

integrity_toolkit usage
integrity_toolkit [OPTION]...
  -u FILENAME      Ublox 1 config YAML file
  -d               Enables dual Rx Mode
  -b FILENAME      Ublox 2 config YAML file (Also requires -d option)
  -r FILENAME      USRP config YAML file (if equipped)
  -i FILENAME      integrity config YAML file (if equipped)
  -p PORT          serial port for NMEA output
  -x LCD PORT      serial port for LCD display
  -l               Enables Logging
if options are not specified, then defaults are:
  ublox1 config = ublox1.yaml
  ublox2 config = ublox2.yaml
  usrp config   = usrp.yaml
  integrity config = integrity.yaml
  lcdPort = /dev/ttyACM0
  port = /dev/ttyUSB99
```

The toolkit requires at least 1 compatible Ublox receiver to operate. The 'config' folder contains pre-canned configuration files for each toolkit component. All configuration inputs (except the port argument) were discussed previously in this document. The 'port' argument specifies the serial port that will be used to publish 2 NMEA messages. The 'lcd port' argument specifies an available LCD device (if equipped).

A sample launch command is as follows:

```
./integrity_toolkit -u ../config/ublox1.yaml -d -b ../config/ublox2.yaml -r ../config/usrp.yaml -i ../config/inte
```

## Quantization Calibration for the USRP

In some cases the USRP gain may need to be adjusted in order to provide adequate signal strength to the acquisition check. This process is termed "quantization calibration" as it aims to place the signal within the dynamic range of the USRPs A/D. This manual process involves adjusting the gain of the USRP (found in `usrp.yaml`) until the displayed sample quantization value appears to be in the linear region of the A/D. The display value range is determined by the sample type used. For example, SC8 samples will be between 0 and 181 (signed 8-bit samples combined I/Q magnitude) where SC16 samples will be between 0 and 46,339.

**Note: If adjusting the gain of the USRP does not effectively place the quantization level in the desired range, external attenuation or amplification may be required.**

It must also be noted that the USRP gain cannot currently be changed on the fly, so the toolkit application will have to be restarted after changing the gain in the config file. The following shows a sample output for a "qcal" process

Set the "qcal" parameter to true in the yaml file

```
qcal:true
```

Launch the toolkit application with the appropriate command line string above. Once all devices have been initialized, the terminal window will begin displaying the quantization value as "MaxBufferValue", such as

```
INFO: Channel [0] : MaxBufferValue = 11.4018, Buffer size = 18446744073709551612,
INFO: Channel [0] : MaxBufferValue = 12.0416, Buffer size = 18446744073709551612,
INFO: Channel [0] : MaxBufferValue = 13.0384, Buffer size = 18446744073709551612,
INFO: Channel [0] : MaxBufferValue = 12.1655, Buffer size = 18446744073709551614,
INFO: Channel [0] : MaxBufferValue = 12.7279, Buffer size = 18446744073709551612,
INFO: Channel [0] : MaxBufferValue = 11, Buffer size = 18446744073709551612,
INFO: Channel [0] : MaxBufferValue = 12.3693, Buffer size = 18446744073709551612,
INFO: Channel [0] : MaxBufferValue = 12.8062, Buffer size = 18446744073709551612,
```

In this particular instance, the gain is set to maximum (75) and the value appears safely in the dynamic range of the USRP. It is suggested that "head room" be allowed in the dynamic range for increasing power levels from interference signals, especially in test environments. Once the "qcal" process is complete, return the parameter to the "false" setting to allow the acquisition check to begin its operation.

## Acquisition Check Tuning

Once the qcal process is complete, it will likely be necessary to tune the Acquisition Check. If possible, this should be done when the user is confident that the antennas are not subject to interfering signals, both intentional and unintentional. The figure below shows a properly tuned Acquisition Check. The parameter "peak\_ratio\_threshold" (acq←\_peak\_ratio\_thresh in plot) should be adjusted so that it sits just above the peak ratio value "noise floor". The "high\_power\_threshold" (acq\_hi\_pwr\_thresh in plot) should be set high enough above the acquisition peaks to allow for signal strength variations but low enough to detect any increases in power that would indicate jamming or spoofing.



```
# =====Acquisition check parameters=====
enable_acquisition_check: true
acq_chek_name: acq_check
if_data_sampling_rate: 5000000
if_data_intermediate_freq: 0.0
acq_check_search_band: 10000.0
acq_check_search_step_size: 500.0
acq_check_integration_period: 0.001
acq_check_code_freq_basis: 1023000.0
acq_check_code_length: 1023
high_power_threshold: 10000000
peak_ratio_threshold: 3.0
acquisition_threshold: 1000000
acq_check_level_period: 5.0
acq_check_inconsistent_thresh: 2.0
acq_check_unassured_thresh: 4.0
acq_check_weight: 1.0
```

## Output Messages

The toolkit will publish two separate NMEA messages at the rate that PVT data is received from the primary Ublox. The first is a custom defined message with the following format

```
$GPSAR,[system timestamp],[assurance_level]
```

i.e.

```
$GPSAR,1608244308.96,3:y
```

The second published string is a standard NMEA GPGGA string, i.e.

```
$GPGGA,223149.00,3232.2271,N,8531.1766,W,3,08,,139.52,M,109.22,M,0.0,0.0
```

The output port is specified as an input argument to the toolkit application. The baud rate is fixed at 9600.

## **Application Window**

Refer to the toolkit user interface guide for a detailed description of the interface window and its components.





## Chapter 2

# Licenses

This library contains 4 separate licenses. Each component is listed below with the corresponding license(s).

- GNU General Public License (usrp\_utilities, integrity\_ui)
- BSD 3-Clause License (pnt\_integrity, ublox, if\_data\_utils, is4s\_common)

The serial library comes with 2 licenses (The MIT License and The Google V8 Project). All license statements are included below for reference.

### GNU GENERAL PUBLIC LICENSE

Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <https://fsf.org/>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

#### Preamble

The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program—to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

## TERMS AND CONDITIONS

### 0. Definitions.

"This License" refers to version 3 of the GNU General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees" and "recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

### **1. Source Code.**

The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work.

A "Standard Interface" means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The "System Libraries" of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A "Major Component", in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The "Corresponding Source" for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

### **2. Basic Permissions.**

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

### **3. Protecting Users' Legal Rights From Anti-Circumvention Law.**

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

#### 4. Conveying Verbatim Copies.

You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

#### 5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

- a) The work must carry prominent notices stating that you modified it, and giving a relevant date.
- b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to "keep intact all notices".
- c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.
- d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

#### 6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

- a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.
- b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.

- c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.
- d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.
- e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A "User Product" is either (1) a "consumer product", which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, "normally used" refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

"Installation Information" for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

## 7. Additional Terms.

"Additional permissions" are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

- a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or
- b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or
- c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or
- d) Limiting the use for publicity purposes of names of licensors or authors of the material; or
- e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or
- f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered "further restrictions" within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

## 8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

#### **9. Acceptance Not Required for Having Copies.**

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

#### **10. Automatic Licensing of Downstream Recipients.**

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An "entity transaction" is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

#### **11. Patents.**

A "contributor" is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's "contributor version".

A contributor's "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

#### **12. No Surrender of Others' Freedom.**

If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

#### **13. Use with the GNU Affero General Public License.**

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

#### **14. Revised Versions of this License.**

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.



#### 15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

#### 16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

#### 17. Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS

#### How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>
```

```
This program is free software: you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation, either version 3 of the License, or
(at your option) any later version.
```

```
This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.
```

```
You should have received a copy of the GNU General Public License
along with this program. If not, see <https://www.gnu.org/licenses/>.
```

Also add information on how to contact you by electronic and paper mail.

If the program does terminal interaction, make it output a short notice like this when it starts in an interactive mode:

```
<program> Copyright (C) <year> <name of author>
This program comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type `show c' for details.
```

The hypothetical commands 'show w' and 'show c' should show the appropriate parts of the General Public License. Of course, your program's commands might be different; for a GUI interface, you would use an "about box".

You should also get your employer (if you work as a programmer) or school, if any, to sign a "copyright disclaimer" for the program, if necessary. For more information on this, and how to apply and follow the GNU GPL, see <https://www.gnu.org/licenses/>.

The GNU General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License. But first, please read <https://www.gnu.org/licenses/why-not-lgpl.html>.

## BSD 3-Clause License

This library is licensed under the . The library contains source code developed by IS4S and third parties. Refer to the individual source files for applicable copyright information.

Copyright (c) 2020 Integrated Solutions for Systems (IS4S), Inc Copyright (c) 2017, ETHZ ASL (geodetic converter)  
Copyright (c) 2009, Ben Hoyt (inih)

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of IS4S, Ben Hoyt or ETHZ ASL nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

## The MIT License

Copyright (C) 2020 Integrated Solutions for Systems, Inc

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

## V8 Google Project (for v8stdint.h)

This header is from the v8 google project: <http://code.google.com/p/v8/source/browse/trunk/include/v8stdint.h>

Copyright 2012 the V8 project authors. All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission. THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.



## Chapter 3

# Build Instructions for the PNT Integrity Toolkit

This repository contains the top level CMake project for building the IS4S PNT Integrity Library, Toolkit and their dependencies.

### System Requirements

The PNT Integrity Toolkit has been tested with on the following operating systems:

- \* Ubuntu Linux 18.04 & 20.04
- \* MacOS 10.15

The following additional tools are needed to build the library:

- \* CMake 3.5 or greater
- \* C++14 compliant compiler (e.g. Clang 3.3+, GCC 4.7+, MSVC 2015+)

### Dependencies

The PNT Integrity Library is designed to require as few third party dependencies as possible to support building on a wide variety of platforms. Two dependencies are required in addition to the libraries provided in the package. The **Eigen** C++ template library for linear algebra is required by the base PNT Integrity library. The **FFTW** package is optionally required and is needed to use the acquisition check. **QT** is required to build and run the user interface

Both packages can be installed following instructions on their respective websites. Eigen is a header-only package and can be installed by downloading a release from the project web site and extracting to a local folder. FFTW binaries are available for a range of platforms from the project web site.

Alternatively, a package manager can be used to install the dependencies. For MacOS the **Homebrew** package manager is recommended. The **Chocolatey** package manager is recommended for Windows. Instructions on installing the required and optional dependencies using package managers on the supported operating systems are provided in the following sections.

## **Ubuntu / Debian**

Install Boost by running:

```
sudo apt-get install libboost-all-dev
```

Install Eigen by running:

```
sudo apt install libeigen3-dev
```

Optionally install FFTW by running:

```
sudo apt install libfftw3-dev
```

Install Qt5 on Ubuntu by running:

```
sudo apt install qtdeclarative5-dev qtwebengine5-dev libqt5charts5-dev
```

Install YAML-CPP by running:

```
sudo apt install libyaml-cpp-dev
```

Install UHD by running:

```
sudo apt-get install libuhd-dev libuhd003 uhd-host
```

## **MacOS (for PNT Integrity Library only)**

Install boost by running:

```
brew install boost
```

Install Eigen by running:

```
brew install eigen
```

Optionally install FFTW by running:

```
brew install fftw
```

Install Qt5 on MacOS by running:

```
brew install qt
```

If you encounter CMake build errors when finding QT this may help:

```
export CMAKE_PREFIX_PATH=/usr/local/Cellar/qt/[version]/
```

Install YAML-CPP by running:

```
brew install yaml-cpp
```

Follow the Ettus [instructions](#) to install UHD.

## Unpackaging

***Skip this step if cloning directly from GitHub / Gitlab***

Extract the release archive:

```
unzip release.zip
cd release
```

## Building

Building can be performed either with a script or manually with commands.

### Script

Execute the provided build script to build

```
./build.sh
```

### Build Commands

Generate build files using cmake

```
mkdir build
```

To build only the PNT Integrity library

```
cmake ../ -DCMAKE_BUILD_TYPE=Release -DBUILD_KIT=FALSE
```

To build the full toolkit with sample application and hardware drivers

```
cmake ../ -DCMAKE_BUILD_TYPE=Release -DBUILD_KIT=TRUE
```

By default, this will generate Unix Makefiles for the package. Project files can be generated for other build systems or IDEs by selecting an alternative **CMake generator**.

Build the libraries by running:

```
make
```

## Installing

The libraries can be optionally installed to the user's system by running:

```
make install
```





## Chapter 4

# Namespace Index

### 4.1 Namespace List

Here is a list of all documented namespaces with brief descriptions:

<a href="#">csv_write</a>	Namespace for the csv writer functions . . . . .	<a href="#">37</a>
<a href="#">integrity_toolkit</a>	Namespace for the toolkit application . . . . .	<a href="#">39</a>



## Chapter 5

# Class Index

### 5.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

<a href="#">csv_write::MsgCsvWrite</a>	
CSV Writer class . . . . .	41
<a href="#">integrity_toolkit::ToolkitApplication</a>	
Container class for all toolkit functionality . . . . .	45
<a href="#">integrity_toolkit::ToolkitLCDDisplay</a> . . . . .	45



## Chapter 6

# File Index

### 6.1 File List

Here is a list of all documented files with brief descriptions:

include/integrity_toolkit/ <a href="#">CsvWrite.hpp</a>	
Class for writing assurance data to csv files . . . . .	49
include/integrity_toolkit/ <a href="#">ToolkitApplication.hpp</a>	
Class definition for main toolkit application . . . . .	50
include/integrity_toolkit/ <a href="#">ToolkitLCDDisplay.hpp</a>	
Class definition for main toolkit application . . . . .	52
src/ <b>CsvWrite.cpp</b> . . . . .	??
src/ <b>ToolkitApplication.cpp</b> . . . . .	??
src/ <b>ToolkitLCDDisplay.cpp</b> . . . . .	??
src/ <a href="#">toolkitMainApplication.cpp</a>	
Main for toolkit application . . . . .	53



## Chapter 7

# Namespace Documentation

### 7.1 csv\_write Namespace Reference

Namespace for the csv writer functions.

#### Classes

- class [MsgCsvWrite](#)  
*CSV Writer class.*

#### Functions

- const std::string [AssuranceReportHeaderString](#) ("timestamp, seq\_num, arrival\_sec, arrival\_nsec, arrival\_timecode, valid\_sec, " "valid\_nsec, valid\_timecode, device\_id, assurance\_level, assurance\_score")  
*Header string for Assurance Report log file.*
- const std::string [AssuranceReportsHeaderString](#) ("timestamp, seq\_num, arrival\_sec, arrival\_nsec, arrival\_timecode, valid\_sec, " "valid\_nsec, valid\_timecode, device\_id, num\_states, [state\_name, weight, " "assurance\_level]")  
*Header string for Assurance Reports log file.*
- std::string [toString](#) (pnt\_integrity::data::AssuranceState msg)  
*Conversion function for AssuranceState.*
- std::string [toString](#) (pnt\_integrity::data::AssuranceReport msg)  
*Conversion function for AssuranceReport.*
- std::string [toString](#) (pnt\_integrity::data::Header msg)  
*Conversion function for Header.*
- std::string [toString](#) (pnt\_integrity::data::AssuranceReports msg)  
*Conversion function for AssuranceReports.*
- std::string [toString](#) (pnt\_integrity::data::AssuranceLevel val)  
*Conversion function for AssuranceLevel.*
- void **findAndReplaceAll** (std::string &data, std::string toSearch, std::string replaceStr)

### 7.1.1 Detailed Description

Namespace for the csv writer functions.

### 7.1.2 Function Documentation

#### 7.1.2.1 toString() [1/5]

```
std::string csv_write::toString (
    pnt_integrity::data::AssuranceState msg )
```

Conversion function for AssuranceState.

##### Parameters

<i>msg</i>	AssuranceState for conversion
------------	-------------------------------

Definition at line 146 of file CsvWrite.cpp.

#### 7.1.2.2 toString() [2/5]

```
std::string csv_write::toString (
    pnt_integrity::data::AssuranceReport msg )
```

Conversion function for AssuranceReport.

##### Parameters

<i>msg</i>	AssuranceReport for conversion
------------	--------------------------------

Definition at line 155 of file CsvWrite.cpp.

#### 7.1.2.3 toString() [3/5]

```
std::string csv_write::toString (
    pnt_integrity::data::Header msg )
```

Conversion function for Header.



**Parameters**

<i>msg</i>	Header for conversion
------------	-----------------------

Definition at line 165 of file CsvWrite.cpp.

**7.1.2.4 toString()** [4/5]

```
std::string csv_write::toString (
    pnt_integrity::data::AssuranceReports msg )
```

Conversion function for AssuranceReports.

**Parameters**

<i>msg</i>	AssuranceReports for conversion
------------	---------------------------------

Definition at line 177 of file CsvWrite.cpp.

**7.1.2.5 toString()** [5/5]

```
std::string csv_write::toString (
    pnt_integrity::data::AssuranceLevel val )
```

Conversion function for AssuranceLevel.

**Parameters**

<i>val</i>	AssuranceLevel for conversion
------------	-------------------------------

Definition at line 192 of file CsvWrite.cpp.

**7.2 integrity\_toolkit Namespace Reference**

Namespace for the toolkit application.

**Classes**

- class [ToolkitApplication](#)  
*Container class for all toolkit functionality.*
- class [ToolkitLCDDisplay](#)

## Variables

- const double `pi` = 3.1415926535898  
*Pi (pi), dimensionless.*
- const double `piOver2` = 0.5 \* `pi`  
*pi/2, dimensionless*
- const double `piTimes2` = 2.0 \* `pi`  
*2\*pi, dimensionless*
- const double `deg2rad` = `pi` / 180.0  
*Ratio of degrees to radians, dimensionless.*
- const double `deg2radSquared` = `deg2rad` \* `deg2rad`  
*Ratio of degrees to radians squared, dimensionless.*
- const double `rad2deg` = 1.0 / `deg2rad`  
*Ratio of radians to degrees, dimensionless.*
- const double `rad2degSquared` = `rad2deg` \* `rad2deg`  
*Ratio of radians to degrees squared, dimensionless.*

### 7.2.1 Detailed Description

Namespace for the toolkit application.

## Chapter 8

# Class Documentation

### 8.1 csv\_write::MsgCsvWrite Class Reference

CSV Writer class.

```
#include <CsvWrite.hpp>
```

#### Public Member Functions

- [MsgCsvWrite](#) ()  
*Construct for writer object.*
- [~MsgCsvWrite](#) ()  
*Destructor for writer object.*
- bool [openFile](#) (std::string relPath, std::string baseFilename, bool appendDate=true)  
*Open the file for writing.*
- bool [closeFile](#) ()  
*Close the CSV file.*
- std::string [getFilename](#) ()  
*Gets the name of the CSV file.*
- bool [isFileOpen](#) ()  
*Indicates if the CSV file has been opened.*
- void [setTimestampsEnabled](#) (bool enabled)  
*Enable or disable time stamping of log file lines.*
- template<class Outputmsg >  
void **logOutput** (Outputmsg msg)
- void [setVerboseOutput](#) (bool verbose)  
*Sets whether verbose output should be used.*
- void [writeHeader](#) (const std::string &line)  
*Writes the header string to the file.*
- const std::string **getCurrentDateTime** ()
- void [writeLine](#) (double timestamp, std::string line)  
*Writes the provided string to the CSV with a timestamp on a unique line.*
- void [writeLine](#) (std::string line)  
*Writes the provided string to the CSV on a unique line.*

### 8.1.1 Detailed Description

CSV Writer class.

Definition at line 79 of file CsvWrite.hpp.

### 8.1.2 Member Function Documentation

#### 8.1.2.1 closeFile()

```
bool csv_write::MsgCsvWrite::closeFile ( )
```

Close the CSV file.

##### Returns

true if the log file is successfully closed

Definition at line 126 of file CsvWrite.cpp.

#### 8.1.2.2 isFileOpen()

```
bool csv_write::MsgCsvWrite::isFileOpen ( ) [inline]
```

Indicates if the CSV file has been opened.

##### Returns

true if the log file is open

Definition at line 121 of file CsvWrite.hpp.

#### 8.1.2.3 openFile()

```
bool csv_write::MsgCsvWrite::openFile (
    std::string relPath,
    std::string baseFilename,
    bool appendDate = true )
```

Open the file for writing.

Creates a CSV file at the specified location with a file name consisting of the base filename plus the current date and time. Format [current\_directory]/[relPath]/[baseFilename]????[extension]

**Parameters**

<i>relPath</i>	Relative path from the working directory to store log file
<i>baseFilename</i>	Base portion of file name before date and time
<i>appendDate</i>	If true, the current data and time is added to the base filename

**Returns**

true if the file is successfully opened

Definition at line 53 of file CsvWrite.cpp.

**8.1.2.4 setTimestampsEnabled()**

```
void csv_write::MsgCsvWrite::setTimestampsEnabled (
    bool enabled ) [inline]
```

Enable or disable time stamping of log file lines.

**Parameters**

<i>enabled</i>	If true, log file lines are timestamped
----------------	---

Definition at line 125 of file CsvWrite.hpp.

**8.1.2.5 setVerboseOutput()**

```
void csv_write::MsgCsvWrite::setVerboseOutput (
    bool verbose )
```

Sets whether verbose output should be used.

**Parameters**

<i>verbose</i>	set to true for additional status data to print to stdout
----------------	---

Definition at line 253 of file CsvWrite.cpp.

#### 8.1.2.6 writeHeader()

```
void csv_write::MsgCsvWrite::writeHeader (
    const std::string & line )
```

Writes the heaer string to the file.

Writer Header to 1st line of CSV file.

##### Parameters

<i>line</i>	The header string to write
-------------	----------------------------

Definition at line 233 of file CsvWrite.cpp.

#### 8.1.2.7 writeLine() [1/2]

```
void csv_write::MsgCsvWrite::writeLine (
    double timestamp,
    std::string line )
```

Writes the provied string to the CSV with a timestamp on a unique line.

##### Parameters

<i>timestamp</i>	The timestamp associated with the log entry
<i>line</i>	The string for the log entry

Definition at line 264 of file CsvWrite.cpp.

#### 8.1.2.8 writeLine() [2/2]

```
void csv_write::MsgCsvWrite::writeLine (
    std::string line )
```

Writes the provied string to the CSV on a unique line.

##### Parameters

<i>line</i>	The string for the log entry
-------------	------------------------------

Definition at line 258 of file CsvWrite.cpp.

The documentation for this class was generated from the following files:

- [include/integrity\\_toolkit/CsvWrite.hpp](#)
- [src/CsvWrite.cpp](#)

## 8.2 integrity\_toolkit::ToolkitApplication Class Reference

Container class for all toolkit functionality.

```
#include <ToolkitApplication.hpp>
```

### Public Member Functions

- [ToolkitApplication](#) (int argc, char \*argv[])  
*Constructor for the application object.*
- [~ToolkitApplication](#) ()  
*Default constructor for the application object.*
- int [run](#) ()  
*Launches the application.*

### 8.2.1 Detailed Description

Container class for all toolkit functionality.

The [ToolkitApplication](#) class is a wrapper container for all of the hardware drivers, libraries, and user interface objects needed for the toolkit application

Definition at line 95 of file [ToolkitApplication.hpp](#).

The documentation for this class was generated from the following files:

- [include/integrity\\_toolkit/ToolkitApplication.hpp](#)
- [src/ToolkitApplication.cpp](#)

## 8.3 integrity\_toolkit::ToolkitLCDDisplay Class Reference

```
#include <ToolkitLCDDisplay.hpp>
```

## Public Member Functions

- [ToolkitLCDDisplay](#) (const std::string &port, int baudRate)  
*Construct for the LCD object.*
- [ToolkitLCDDisplay](#) ()  
*Default constructor for the LCD Object.*
- bool [initialize](#) (const std::string &port, int baudRate)  
*Initializes the LCD display with provided port and baud rate.*
- void [SetAssuranceLevel](#) (const pnt\_integrity::data::AssuranceLevel &level)  
*Sets and displays the total assurance level on the display.*
- void [displayCheckState](#) (pnt\_integrity::data::AssuranceState &state)  
*Displays the assurance level of a particular state lower on the display.*
- [~ToolkitLCDDisplay](#) ()  
*Destructor for the LCD display object.*
- void [clearDisplay](#) ()  
*Clears display.*
- void [clearLeds](#) ()  
*Clears LEDs.*

### 8.3.1 Detailed Description

Container class for interacting with the Crystal Fontz LCD display in the toolkit

Definition at line 36 of file ToolkitLCDDisplay.hpp.

### 8.3.2 Constructor & Destructor Documentation

#### 8.3.2.1 ToolkitLCDDisplay()

```
ToolkitLCDDisplay::ToolkitLCDDisplay (
    const std::string & port,
    int baudRate )
```

Construct for the LCD object.

#### Parameters

<i>port</i>	The serial port for the device
<i>baudRate</i>	The baud rate for serial comms with the device

Definition at line 30 of file ToolkitLCDDisplay.cpp.



### 8.3.3 Member Function Documentation

#### 8.3.3.1 displayCheckState()

```
void ToolkitLCDDisplay::displayCheckState (
    pnt_integrity::data::AssuranceState & state )
```

Displays the assurance level of a particular state lower on the display.

##### Parameters

<i>state</i>	The provided Assurance State
--------------	------------------------------

Definition at line 88 of file ToolkitLCDDisplay.cpp.

#### 8.3.3.2 initialize()

```
bool ToolkitLCDDisplay::initialize (
    const std::string & port,
    int baudRate )
```

Initializes the LCD display with provided port and baud rate.

##### Parameters

<i>port</i>	The serial port for the device
<i>baudRate</i>	The baud rate for serial comms with the device

Definition at line 36 of file ToolkitLCDDisplay.cpp.

#### 8.3.3.3 SetAssuranceLevel()

```
void ToolkitLCDDisplay::SetAssuranceLevel (
    const pnt_integrity::data::AssuranceLevel & level )
```

Sets and displays the total assurance level on the display.

**Parameters**

<i>level</i>	The current total assurance level
--------------	-----------------------------------

Definition at line 45 of file ToolkitLCDDisplay.cpp.

The documentation for this class was generated from the following files:

- include/integrity\_toolkit/[ToolkitLCDDisplay.hpp](#)
- src/ToolkitLCDDisplay.cpp

## Chapter 9

# File Documentation

### 9.1 include/integrity\_toolkit/CsvWrite.hpp File Reference

Class for writing assurance data to csv files.

```
#include <algorithm>
#include <atomic>
#include <fstream>
#include <iostream>
#include <string>
#include <vector>
#include "cf_display/cf_display.h"
#include "cf_display/typedefs.h"
#include "integrity_toolkit/ToolkitLCDDisplay.hpp"
#include "logutils/logutils.hpp"
#include "pnt_integrity/AssuranceCheck.hpp"
#include "pnt_integrity/IntegrityData.hpp"
#include "pnt_integrity/IntegrityMonitor.hpp"
```

#### Classes

- class [csv\\_write::MsgCsvWrite](#)  
*CSV Writer class.*

#### Namespaces

- [csv\\_write](#)  
*Namespace for the csv writer functions.*

## Functions

- const std::string [csv\\_write::AssuranceReportHeaderString](#) ("timestamp, seq\_num, arrival\_sec, arrival\_nsec, arrival\_timecode, valid\_sec, " "valid\_nsec, valid\_timecode, device\_id, assurance\_level, assurance\_score")  
*Header string for Assurance Report log file.*
- const std::string [csv\\_write::AssuranceReportsHeaderString](#) ("timestamp, seq\_num, arrival\_sec, arrival\_nsec, arrival\_timecode, valid\_sec, " "valid\_nsec, valid\_timecode, device\_id, num\_states, [state\_name, weight, " "assurance\_level]")  
*Header string for Assurance Reports log file.*
- std::string [csv\\_write::toString](#) (pnt\_integrity::data::AssuranceState msg)  
*Conversion function for AssuranceState.*
- std::string [csv\\_write::toString](#) (pnt\_integrity::data::AssuranceReport msg)  
*Conversion function for AssuranceReport.*
- std::string [csv\\_write::toString](#) (pnt\_integrity::data::Header msg)  
*Conversion function for Header.*
- std::string [csv\\_write::toString](#) (pnt\_integrity::data::AssuranceReports msg)  
*Conversion function for AssuranceReports.*
- std::string [csv\\_write::toString](#) (pnt\_integrity::data::AssuranceLevel val)  
*Conversion function for AssuranceLevel.*

### 9.1.1 Detailed Description

Class for writing assurance data to csv files.

#### Author

Anna Levasseur  
[josh.clanton@is4s.com](mailto:josh.clanton@is4s.com)

#### Date

2/22/2021

## 9.2 include/integrity\_toolkit/ToolkitApplication.hpp File Reference

Class definition for main toolkit application.

```
#include <QApplication>
#include <chrono>
#include <fstream>
#include <future>
#include <limits>
#include <mutex>
#include <string>
#include <thread>
#include <uhd/transport/bounded_buffer.hpp>
#include <utility>
```

```

#include "if_data_utils/IFDataFileReader.hpp"
#include "if_data_utils/IFSampleData.hpp"
#include "integrity_toolkit/ToolkitLCDDisplay.hpp"
#include "integrity_ui_base/mainwindow.h"
#include "logutils/logutils.hpp"
#include "pnt_integrity/AgcCheck.hpp"
#include "pnt_integrity/AngleOfArrivalCheck.hpp"
#include "pnt_integrity/ClockBiasCheck.hpp"
#include "pnt_integrity/CnoCheck.hpp"
#include "pnt_integrity/IntegrityData.hpp"
#include "pnt_integrity/IntegrityMonitor.hpp"
#include "pnt_integrity/NavigationDataCheck.hpp"
#include "pnt_integrity/PositionJumpCheck.hpp"
#include "pnt_integrity/PositionVelocityConsistencyCheck.hpp"
#include "pnt_integrity/RangePositionCheck.hpp"
#include "pnt_integrity/StaticPositionCheck.hpp"
#include "serial/serial.h"
#include "ublox/ublox.h"
#include "usrp_utilities/SurpasUsrpDevice.hpp"
#include "yaml_parser/yaml_parser.hpp"
#include "integrity_toolkit/CsvWrite.hpp"

```

## Classes

- class [integrity\\_toolkit::ToolkitApplication](#)  
*Container class for all toolkit functionality.*

## Namespaces

- [integrity\\_toolkit](#)  
*Namespace for the toolkit application.*

## Variables

- const double [integrity\\_toolkit::pi](#) = 3.1415926535898  
*Pi (pi), dimensionless.*
- const double [integrity\\_toolkit::piOver2](#) = 0.5 \* pi  
*pi/2, dimensionless*
- const double [integrity\\_toolkit::piTimes2](#) = 2.0 \* pi  
*2\*pi, dimensionless*
- const double [integrity\\_toolkit::deg2rad](#) = pi / 180.0  
*Ratio of degrees to radians, dimensionless.*
- const double [integrity\\_toolkit::deg2radSquared](#) = deg2rad \* deg2rad  
*Ratio of degrees to radians squared, dimensionless.*
- const double [integrity\\_toolkit::rad2deg](#) = 1.0 / deg2rad  
*Ratio of radians to degrees, dimensionless.*
- const double [integrity\\_toolkit::rad2degSquared](#) = rad2deg \* rad2deg  
*Ratio of radians to degrees squared, dimensionless.*

### 9.2.1 Detailed Description

Class definition for main toolkit application.

**Author**

Josh Clanton [josh.clanton@is4s.com](mailto:josh.clanton@is4s.com)

**Date**

April 20, 2020

## 9.3 include/integrity\_toolkit/ToolkitLCDDisplay.hpp File Reference

Class definition for main toolkit application.

```
#include "cf_display/cf_display.h"
#include "cf_display/typedefs.h"
#include "logutils/logutils.hpp"
#include "pnt_integrity/IntegrityData.hpp"
```

**Classes**

- class [integrity\\_toolkit::ToolkitLCDDisplay](#)

**Namespaces**

- [integrity\\_toolkit](#)  
*Namespace for the toolkit application.*

### 9.3.1 Detailed Description

Class definition for main toolkit application.

**Author**

Josh Clanton [josh.clanton@is4s.com](mailto:josh.clanton@is4s.com)

**Date**

April 20, 2021

## 9.4 src/toolkitMainApplication.cpp File Reference

Main for toolkit application.

```
#include "integrity_toolkit/ToolkitApplication.hpp"
```

### Functions

- `int main (int argc, char *argv[])`  
*The main function for the toolkit application.*

### 9.4.1 Detailed Description

Main for toolkit application.

#### Author

Josh Clanton [josh.clanton@is4s.com](mailto:josh.clanton@is4s.com)

#### Date

April 20, 2020





# Index

closeFile  
     csv\_write::MsgCsvWrite, [42](#)  
 csv\_write, [37](#)  
     toString, [38](#), [39](#)  
 csv\_write::MsgCsvWrite, [41](#)  
     closeFile, [42](#)  
     isFileOpen, [42](#)  
     openFile, [42](#)  
     setTimestampsEnabled, [43](#)  
     setVerboseOutput, [43](#)  
     writeHeader, [43](#)  
     writeLine, [44](#)  
  
 displayCheckState  
     integrity\_toolkit::ToolkitLCDDisplay, [47](#)  
  
 include/integrity\_toolkit/CsvWrite.hpp, [49](#)  
 include/integrity\_toolkit/ToolkitApplication.hpp, [50](#)  
 include/integrity\_toolkit/ToolkitLCDDisplay.hpp, [52](#)  
 initialize  
     integrity\_toolkit::ToolkitLCDDisplay, [47](#)  
 integrity\_toolkit, [39](#)  
 integrity\_toolkit::ToolkitApplication, [45](#)  
 integrity\_toolkit::ToolkitLCDDisplay, [45](#)  
     displayCheckState, [47](#)  
     initialize, [47](#)  
     SetAssuranceLevel, [47](#)  
     ToolkitLCDDisplay, [46](#)  
 isFileOpen  
     csv\_write::MsgCsvWrite, [42](#)  
  
 openFile  
     csv\_write::MsgCsvWrite, [42](#)  
  
 SetAssuranceLevel  
     integrity\_toolkit::ToolkitLCDDisplay, [47](#)  
 setTimestampsEnabled  
     csv\_write::MsgCsvWrite, [43](#)  
 setVerboseOutput  
     csv\_write::MsgCsvWrite, [43](#)  
 src/toolkitMainApplication.cpp, [53](#)  
  
 toString  
     csv\_write, [38](#), [39](#)  
 ToolkitLCDDisplay  
     integrity\_toolkit::ToolkitLCDDisplay, [46](#)  
  
 writeHeader  
     csv\_write::MsgCsvWrite, [43](#)  
 writeLine  
     csv\_write::MsgCsvWrite, [44](#)