User Manual and Guide
OPAL™ Performance Series
3D LiDAR Scanner

Revision 6 March 2018
**NOTICE**

This document contains Neptec Technologies Corp. proprietary information. Use, disclosure, copying or distribution of any part of the information contained herein, beyond that for which it was originally furnished, requires the written permission of Neptec Technologies Corp.

The information in this document is subject to change without notice and relates only the product defined in the introduction of this document. Neptec Technologies Corp. intends that information contained herein is, to the best of its knowledge, correct and accurate. However, any/all liabilities associated with the use or accuracy of the information contained herein must be defined in a separate agreement between Neptec Technologies Corp. and the customer/user.

*Copyright © Neptec Technologies Corp. 2018.*

*All rights reserved.*
RADIO AND TELEVISION INTERFERENCE

**NOTE:** This equipment has been tested and found to comply with the limits for a Class A digital device, pursuant to Part 15 of the FCC rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment.

This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference in which case the user will be required to correct the interference at their expense.

In order to maintain compliance with FCC regulations shielded cables must be used with this equipment. Operation with non-approved equipment or unshielded cables is likely to result in interference to radio & television reception.
Table of Contents

1.0 Introduction .................................................................................................................. 7
1.1 Purpose .......................................................................................................................... 7
1.2 Audience ......................................................................................................................... 7
1.3 Associated Publications ................................................................................................. 7
1.4 Symbols and Conventions .............................................................................................. 8
1.5 Safety Guidelines .......................................................................................................... 8
1.6 Laser Radiation .............................................................................................................. 9
  1.6.1 Laser Output Aperture ............................................................................................... 10
  1.6.2 Digital Cameras ........................................................................................................ 11
2.0 OPAL Features ............................................................................................................... 12
  2.1 Applications .................................................................................................................. 12
  2.2 3DRi System Manager Software .................................................................................. 12
  2.3 Physical Features ........................................................................................................ 12
  2.4 Included Items ............................................................................................................. 13
  2.5 Technical Specifications ............................................................................................. 14
3.0 Theory of Operation ..................................................................................................... 15
  3.1 Time-of-flight LiDAR ................................................................................................... 15
  3.2 Obscurant Penetration ................................................................................................. 15
  3.3 Scan Pattern ............................................................................................................... 16
4.0 Installation ..................................................................................................................... 17
  4.1 OPAL Mounting Requirements .................................................................................... 17
  4.2 Mounting Methods ....................................................................................................... 17
    4.2.1 Primary Mounting Interface .................................................................................... 17
    4.2.2 Additional Mounting Interface ................................................................................ 18
  4.3 Cabling Requirements ................................................................................................. 19
    4.3.1 Chassis Ground ....................................................................................................... 19
    4.3.2 Cable Routing ......................................................................................................... 19
    4.3.3 Surge Protection ..................................................................................................... 19
    4.3.4 Cable Length .......................................................................................................... 20
  4.4 Hardware Installation ................................................................................................... 20
5.0 3DRi Software .............................................................................................................. 21
  5.1 Quick Start ................................................................................................................... 21
  5.2 Web Interface .............................................................................................................. 21
    5.2.1 Configuring OPAL Network Settings ....................................................................... 22
    5.2.2 Configuring OPAL Parameters through the Web Interface .................................. 24
### Table of Contents

5.3 3DRi Software Installation

5.3.1 Installing on Windows 7/10

5.3.2 Installing on Ubuntu Linux 14.04

5.3.3 Obtaining a License Key

5.3.4 Upgrading Your Software

5.4 3DRi Software Architecture

5.5 3DRi Plug-ins

5.6 3DRi Register Plugin

5.6.2 OPAL Sensor Coordinate Conventions

5.6.3 User Navigation App Requirements

5.6.4 Configure 3DRi Software for Moving Platform

5.6.5 OPAL at a Known Fixed GPS Location

5.7 3DRi Obscurants Plugins

5.7.2 Configuring Obscurants

6.0 Operating the OPAL Sensor

6.1 Start the OPAL Sensor

6.2 Start 3DRi SystemManager (Optional)

6.3 Start the 3DRi Viewer

6.4 Select a Scan Pattern

6.5 Customizing the Scan Parameters

6.6 Performing a Scan

6.7 Using the Workspace Panel

6.8 Saving the Scan Data

6.9 Displaying the Scan Data

6.10 Using the Mouse

6.10.1 Zoom, Rotate and Shift the Data

6.10.2 Perform a Measurement Between Two Points

6.11 Displaying the List of 3D Controls

6.12 Displaying the Reference Frame Axes

6.13 Viewing the Data Along a Specified Coordinate Axis

6.14 Viewing the Data Within a Specified Area of Interest

6.15 Viewing the Scan Data in Radar Plot Mode

6.16 Using the Display Properties Panel

6.16.1 The Apply Button

6.16.2 Selecting a Dataset

6.16.3 Selecting the Type
1.0 Introduction

This document is intended as a comprehensive user manual for Neptec Technologies’ third generation of OPAL™ family of obscurant penetrating 3D LiDAR.

1.1 Purpose

The purpose of this user manual is to provide technical personnel with the information necessary to carry out:

- secure mounting
- electrical installation
- system configuration
- system operation
- maintenance

1.2 Audience

<table>
<thead>
<tr>
<th>Activity</th>
<th>Target Group</th>
</tr>
</thead>
<tbody>
<tr>
<td>Installation</td>
<td>Service Engineers</td>
</tr>
<tr>
<td>Maintenance</td>
<td>Service Technicians</td>
</tr>
<tr>
<td>Replacement</td>
<td></td>
</tr>
<tr>
<td>Configuration</td>
<td>Operations Engineers</td>
</tr>
<tr>
<td>Operation</td>
<td>Operations Technicians</td>
</tr>
<tr>
<td>Troubleshooting</td>
<td></td>
</tr>
</tbody>
</table>

1.3 Associated Publications

- OPAL3 Laser Safety Analysis (Neptec Technologies Corp. document 80-10207)
1.4 Symbols and Conventions

This symbol indicates that the user should ensure they adhere to the safety instructions listed herein when installing or operating the equipment.

Throughout this manual this symbol indicates a procedure that has the potential for laser beam exposure.

1.5 Safety Guidelines

OPAL systems must be installed, operated, and serviced by trained personnel. Personnel not trained or familiar with the operating characteristics of OPAL systems should not be allowed to operate the equipment.

- Warnings and cautions in this User Manual must be observed.
- When installing, operating, testing, or decommissioning OPAL systems, always follow the national and local codes, standards and regulations.
- The Owners, operators and their representatives of the structure, vehicle or surface on which the OPAL system is installed are responsible for observing all applicable safety regulations and rules.
- The User Manual must be made available to the operator of the structure, vehicle or surface where the OPAL is installed and operating.
- Use of controls, adjustments, or procedures, other than those specified herein, may result in hazardous laser light exposure.
- Follow the installation sequence described in Section 4.4 when installing the external data and the external power harness. Always follow current safety regulations when performing electrical work.
- In the event of servicing, maintenance, or an emergency, remove power from the unit by disconnecting the DC power input connector located on the rear of the unit. By rotating the connector in a counter-clockwise direction the connector will disconnect from the unit.
1.6 Laser Radiation

The OPAL Sensor is Class 1 (eye safe) as per IEC 60825-1: 2007, i.e. complies with 21CFR1040.10. The laser beam is invisible to the human eye. No corrective maintenance is necessary to ensure compliance with laser Class 1.

Two different labels on the casing (see Figure 1-1) provide information about the scanner:

![Figure 1-1: OPAL Label Locations](image)

![Figure 1-2: Certification Label (Label 1)](image)
Model number, serial number and general sensor description are replaced with sensor-specific numbers and configuration on an OPAL scanner.

1.6.1 Laser Output Aperture

The laser output aperture of the OPAL sensor is located in the optical window, as shown in Figure 1-4. Laser pulses are emitted and received back through the protective optical window. The field of view is model dependent (see Figure 1-4). Follow the recommended mounting guidelines when installing OPAL on a vehicle; refer to Section 4.0.

Do not touch the optical window. Scratches or smearing caused by hand contact may degrade the performance of the OPAL Sensor. Clean using approved materials.
1.6.2 Digital Cameras

**Caution:** The OPAL Sensor uses infrared laser light, which may be damaging to digital cameras and similar electronic devices. To avoid damage, do not point camera equipment at the optical window when the OPAL Sensor is operating.
# 2.0 OPAL Features

The OPAL system is a rugged, multi-purpose, 3D laser sensor, specifically designed for harsh environments. It is a time-of-flight Light Detection and Ranging (LiDAR) system which, using a unique spinning prism architecture, collects, in seconds, 3D point cloud data in a conical or 360° degree azimuth fields of view (depending on model chosen).

Unlike conventional LiDAR scanners, the OPAL is also able to penetrate obscurants such as dust, rain, snow, fog and smoke, using Neptec’s patented obscurant-penetrating LiDAR technology\(^1\).

The sensor is specifically designed to operate in harsh environmental conditions. It has an environmentally sealed (IP65) cast aluminum housing and has no external moving parts. It has a -40°C to +40°C (See section 2.5) operating temperature range and is ruggedized to withstand significant vibration and shock levels.

## 2.1 Applications

The OPAL is designed for rugged autonomous solutions whether on land, sea or air. It can also be used for more conventional survey and measurement tasks that require outdoor installations.

The OPAL series of LiDAR sensors are part of Neptec’s OPAL systems family of 3D sensors. All OPAL sensors are compatible with Neptec’s 3DRi (3D Real-time intelligence) software development kit (SDK) that provides optional features such as real-time change detection and object recognition and tracking. This SDK can be used to easily develop advanced 3D applications. The SDK consists of the 3DRi System Manager and optional 3DRi Plug-ins.

## 2.2 3DRi System Manager Software

All OPAL series sensors include the 3DRi System Manager software. The 3DRi System Manager provides functionality to configure and manually operate OPAL Sensors. It also includes an Application Programming Interface (API). The OPAL data may be easily loaded into various commercially available 3D point cloud processing software packages. The 3DRi System Manager supports various common 3D data formats (ASCII, .pif, etc.).

## 2.3 Physical Features

Figure 2-1 shows the principal features of the OPAL.

\(^1\) An optional software plug-in is required to enable this feature.
2.4 Included Items

An OPAL system is shipped with the following items:

- OPAL 3D LiDAR Sensor
- Power Cable (3 m length, other lengths available)
- Ethernet Data Cable (3 m length, other lengths available)
- Ethernet Data Cable Rugged Termination Kits
- USB Flash Drive with:
  - 3DRi Software Installation Package
  - OPAL User Manual

The OPAL Sensor and associated items typically ships in a custom Pelican case with wheels, packaged in a protective corrugated cardboard box.
### 2.5 Technical Specifications

<table>
<thead>
<tr>
<th>Mechanical</th>
<th>Power</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Dimensions</strong></td>
<td><strong>Supply</strong></td>
</tr>
<tr>
<td>(H x W x D)</td>
<td>18 – 36 V DC</td>
</tr>
<tr>
<td>17.8cm x 17.8cm x 34.5cm (7.0in x 7.0in x 13.6in)</td>
<td><strong>Consumption</strong></td>
</tr>
<tr>
<td>Weight</td>
<td>110 W TYP; 220 W MAX</td>
</tr>
<tr>
<td>12.7 kg (28 lbs)</td>
<td><strong>Connector</strong></td>
</tr>
<tr>
<td>Housing</td>
<td>Circular 3 pole (Supplier: Hirose Electric; Part Number: RM15WTPZ-3S(71))</td>
</tr>
<tr>
<td>Machined Aluminum IP65 rating</td>
<td></td>
</tr>
</tbody>
</table>

### Laser

<table>
<thead>
<tr>
<th>Field of View</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Type</strong></td>
<td><strong>Interfaces</strong></td>
</tr>
<tr>
<td>1.55 µm pulsed - Class 1 (eye safe)</td>
<td>PoE Gigabit Ethernet, PPS input (time sync), USB</td>
</tr>
<tr>
<td>Conical 45° FOV standard (60°, 90° and 120° available upon request)</td>
<td></td>
</tr>
<tr>
<td>Panoramic 360° by 45° (+5° -40°) FOV standard</td>
<td></td>
</tr>
<tr>
<td><strong>Range</strong></td>
<td><strong>Acquisition rate</strong></td>
</tr>
<tr>
<td>Up to 4000m (model dependent)</td>
<td>Up to 300,000 (single return)</td>
</tr>
<tr>
<td><strong>Penetrable obscurants</strong></td>
<td><strong>Output</strong></td>
</tr>
<tr>
<td>dust, snow, rain, fog and smoke</td>
<td>Time stamped x, y, z with 14 bit intensity</td>
</tr>
<tr>
<td><strong>PoE Connector</strong></td>
<td></td>
</tr>
<tr>
<td>(Supplier: Phoenix Contact; Part Number: 1422205)</td>
<td></td>
</tr>
</tbody>
</table>

### Environmental

| **Operating temperature** | **Humidity** |
| -40°C to +40°C (Consult factory for higher temperature installations) | 100% non-condensing |
| **Altitude** | **Ingress Protection** |
| -1500 ft to +40,000 ft | IP65 |
| **Operational shock** | **Operational Vibration** |
| 5 g half sine 16 ms. | 15Hz to 2kHz, 0.04 g²/Hz |
3.0 Theory of Operation

3.1 Time-of-flight LiDAR

A 3D laser sensor, such as the OPAL, works by emitting a pulse of laser light and detecting the reflection of that pulse in order to accurately determine the distance to the reflected object. The sensor measures the time of flight of the optical pulse to travel to and from the reflected surface. The distance the pulse travelled is then calculated based on the speed of light.

Rather than making a single measurement (as in a laser range finder), 3D laser scanners use rotating mirrors, or other means, to trace a pattern with the laser over a scene, creating millions of range measurements in just a few seconds or minutes. The OPAL Sensor uses spinning prisms to create its scan patterns. By correlating the range measured with the angular position of the laser for each measurement, a 3D point cloud of the scene may be generated. Many laser sensors, including the OPAL, will also output the intensity of the reflected target for each point. This data can be used to distinguish between reflective and non-reflective surfaces at a similar range.

3.2 Obscurant penetration

Conventional 3D laser scanners cannot penetrate obscurants such as dust, rain, snow, fog and smoke, as the obscurant particles are imaged by the scanner instead of the target engulfed in the obscurants. Neptec has developed a patented LiDAR technology that can penetrate significant amounts of obscurants in real-time. This technology was initially developed for LiDAR sensors used on helicopters landing in the desert under brownout conditions. It uses waveform technology to look at the characteristics of the reflected pulse in real-time and distinguish between reflections from obscurants and from the real target.

Neptec has integrated this technology into its OPAL systems 3D laser sensors. Your OPAL Sensor has the ability to detect up to 7 pulses (depending on your model of OPAL). The 3DRi System Manager software (with the optional 3DRi Obscurants plug-in) monitors and filters the 3D data from the sensor for the presence of obscurants. This process happens in real-time and is transparent to the user. You can access both the raw 3D data and the data with the obscurant returns removed through the 3DRi System Manager.

![Figure 3-1: OPAL in a dusty environment](image)
3.3 Scan Pattern

OPAL sensors produce a unique non-overlapping scan pattern (Figure 3-2) that can generate very dense 3D data sets in seconds even when stationary i.e. no data gaps (Figure 3-3). The OPAL Conical can be ordered with a standard 45° degree field of view; 60°, 90° and 120° are also available. The OPAL Panoramic can be ordered with a standard 360° by 45° (+5° -40°) degree field of view. A variety of standard scan patterns for different types of applications are user selectable in the OPAL software interface, or they can be easily customized to alleviate, for example, shadowing effects in mobile or aerial mapping applications.

![Illustration of the OPAL conical scan pattern](image)

Figure 3-2: Illustration of the OPAL conical scan pattern

![OPAL scan pattern after 1 second (left) and 5 seconds (right)](image)

Figure 3-3: OPAL scan pattern after 1 second (left) and 5 seconds (right)

Your 3DRi System Manager includes a number of default scan patterns for different types of applications which can be selected from a drop-down menu. They have been optimized around even point distribution, point density, scan time and the specific application.
4.0 Installation

The sensor is a self-contained, IP65 compliant unit that houses all the electronics and optics needed for scanning.

4.1 OPAL Mounting Requirements

The OPAL should be mounted with no obstructions in the field of view. Whenever possible, minimize the number of close proximity high reflectance targets, such as retro reflectors, in the field of view. Ensure that the mounting location(s) chosen are sufficiently stiff and structurally strong enough to support the OPAL sensor during operations. Installation Control Diagrams for the OPAL and mounting brackets can be found in the Appendix.

4.2 Mounting Methods

4.2.1 Primary Mounting Interface

The primary mounting method, used on all moving platforms and for most other installations, utilizes four (4) 3/8”-24 threaded inserts located on the underside of the OPAL enclosure (see Figure 4-1). A mating mounting bracket (see Figure 4-2) is available, allowing rigid or adjustable angular positioning of the sensor.

Figure 4-1: Main Mounting Holes for the OPAL Sensor
4.2.2 Additional Mounting Interface

The additional mounting interface consists of four 10-24 threaded inserts located on the top of the unit that allow for mounting additional equipment. Clearance must be allowed for all mating connectors when using this additional mounting interface. Do not mount equipment where it could adversely affect thermal dissipation.
4.3 Cabling Requirements

The OPAL sensor uses two cables: a power cable and a data cable (Ethernet). Both use IP65 compliant connectors. Figure 4-4 shows the power cable wiring pin out details. A full cable ICD can be found in the Appendix section of this document. The Ethernet cable terminates one end with a standard RJ45 connector and the other with an IP65 Phoenix Contact (1422205) to connect into the OPAL sensor. You can connect to any of the four (4) OPAL sensor PoE Gigabit Ethernet ports to communicate with the LiDAR. The OPAL sensor also ships with an additional connector kit to terminate an additional cable. Refer to Phoenix Contact installation guide (for connector kit 1422205) to terminate other cables. For applications using PoE peripheral equipment, ensure the cable used to connect to the peripherals complies with the IEEE 802.3 PoE standards.

<table>
<thead>
<tr>
<th>SIGNAL</th>
<th>PWR INPUT</th>
<th>PWR LEADS</th>
<th>WIRE</th>
<th>AWG</th>
</tr>
</thead>
<tbody>
<tr>
<td>PWR_-+</td>
<td>1</td>
<td>FORK LUG</td>
<td>RED</td>
<td>16</td>
</tr>
<tr>
<td>PWR_-</td>
<td>2</td>
<td>FORK LUG</td>
<td>BLK</td>
<td>16</td>
</tr>
<tr>
<td>CHASSIS GROUND</td>
<td>3</td>
<td>FORK LUG</td>
<td>BLUE</td>
<td>16</td>
</tr>
</tbody>
</table>

*Figure 4-4: Power Cable Pin Out*

4.3.1 Chassis Ground

In addition to grounding the power cable, the OPAL should be grounded using the chassis ground connection (Figure 2-1).

4.3.2 Cable Routing

Route cables to minimize run lengths. Secure cables using sturdy mounting clamps or rugged external grade zip ties at regular intervals to prevent undesired motion. Do not route cables where they may be exposed to flying or falling debris.

While the cables are shielded against unwanted electromagnetic effects, routing cables near strong electromagnetic field sources is not recommended.

4.3.3 Surge Protection

The OPAL unit has built-in protection for rugged use and dirty power installations. In situations where higher voltage transients on power supply lines may be expected (such as may be experienced in installations on structures exposed to lightning strikes or other high-voltage fast transients) additional protection is recommended.

The user should consider adding a transient protection device, such as the Transtector DC Defender Model 1101-1110, from Smiths Power, or an equivalent device. Follow the installation and safety requirements in the surge protector manual.
4.3.4 Cable Length

OPAL systems are shipped with 3 m length data and power cables. Custom lengths can be made available on request.

4.4 Hardware Installation

Perform the following steps to install the OPAL system hardware. It is recommended that the installation be performed by a team of two people.

1. Inspect the contents of the OPAL Unit shipment to ensure that all ordered components are present.
2. Review this Installation Planning section thoroughly.
4. Install the desired OPAL mounting bracket/pole in the appropriate location on the vehicle, or structure. Ensure that the bracket/pole and supporting structure are rigid and robust.
5. Mount the OPAL on the mounting bracket using SAE Grade 5 Bolts or better. Ensure that the 3/8”-24 fasteners are properly torqued to 259 in-lbs or 29 N-m.
6. Route and secure cables from the OPAL to the network, and power source. Provide sufficient service loop at source ends to allow for cable repair should the cable become damaged at some future date.
7. Connect the power cable to the power source, fuse panel or other. Do not energize at this time.
8. Verify that the power voltage and polarity measured at the connector end of the power cable harness are as per the wiring diagram. Do not reverse polarity.
9. Verify all connections are correct.
10. Energize the system.
11. Power up the OPAL and perform a scan to verify that the installation was successful.
5.0 3DRi Software

3DRi (3D Real-time intelligence) is a collection of highly efficient algorithms and software to extract actionable information from 3D sensor data in real-time and from moving platforms. This technology, when combined with our OPAL Sensor, allows a machine or system to perceive its environment, automatically recognize changes, and identify and track objects in real-time and while moving. 3DRi makes it easy to develop, integrate and support intelligent 3D machine vision systems for machine automation and other industrial applications.

For a full description of how to install, configure, and operate the 3DRi software modules, please follow through this section.

5.1 Quick Start

After properly mounting and powering the unit following the instructions in Section 4.4, connect to the OPAL using an Ethernet cable plugged into one of the four ports located on the back of the LiDAR. A typical system diagram can be seen in Figure 5-1.

![Figure 5-1: Illustration of a typical OPAL single-sensor setup](image)

Install the provided 3DRi software and launch the 3DRi Viewer. If connected to the OPAL, the viewer will automatically detect the presence of the OPAL on the network using multicast LCM messaging. If you can't see the OPAL, ensure multicast traffic is allowed. Navigate to the 'Sensor Control' pane in the 3DRi Viewer and click "start scan" using the default parameters. A data source will appear in the 'Workspace Pane'. Once the scanner reaches its set motor speeds, LiDAR data will begin to be displayed.

5.2 Web Interface

The OPAL scanner has a web interface that allows for the monitoring, configuration, and control of the LiDAR. The OPAL is supplied with a default IP address (192.168.1.100), which can be changed as required. The computer will need an IP address on the same subnet as the OPAL scanner e.g. 192.168.1.150. Once the computer is configured, use a web browser to navigate to the OPAL’s default IP of 192.168.1.100 and login with the default username and password (admin / admin), the login page can be seen in Figure 5-2.
Once logged in, four panels of information are available; each panel can be displayed by clicking on the title:

- **General Information** – this page shows an overview of diagnostic information such as uptime, system temperature, current status, scanning mode, etc.

- **Status Information** – this page enumerates all available diagnostic information from each subsystem within the OPAL.

- **OPAL Configuration** – this page allows configuration of the startup parameters, such as scanning mode, laser power, and detection mode. This page also allows configuration of the OPAL’s network settings. This is described in detail in Section 5.2.1.

- **System information** – this page displays the current system firmware version, allows exporting diagnostic and system information, and updating of the device firmware.

### 5.2.1 Configuring OPAL Network Settings

Once logged into the web interface (user/password of admin/admin) by navigating in a browser to the OPAL’s default IP of **192.168.1.100**, navigate to the ‘Configuration’ page by clicking the tab’s title bar (Figure 5-3).
Then click on the current IP address, subnet mask, or gateway and change the values to the new desired settings (Figure 5-4). Once complete, click either “Apply Settings” to set the parameters temporarily (until system restart) or click “Save Settings” to maintain changes after system reboot. The interface will prompt indicating that the IP changes have been made and the web browser needs to be re-directed to the new address.

![Network Configuration](image)

**Figure 5-4: Configuring OPAL Network Settings**
5.2.2 Configuring OPAL Parameters through the Web Interface

Several system parameters can be modified on the configuration web interface page. By modifying these parameters then clicking “Apply Settings” the parameter change will take effect immediately but won’t persist after system restart. The parameters can also be saved perpetually after restarts by clicking “Save Settings”. These parameters include:

- Inner Motor Speed* – the rate at which the inner motor rotates (RPS)
- Outer Motor Speed* - the rate at which the outer motor rotates (RPS)
- Laser Power – the percentage of maximum laser power to be emitted (%)
- Scan Duration – the number of seconds for which scanning is to take place. A value of 0 will apply continuous scanning
- Auto Scan – When enabled, the LiDAR will automatically begin to start scanning after a restart using the default scanning parameters
- PRF (Pulse Repetition Frequency) – This sets the pulse rate of the LiDAR (max 300 kHz)
- Detector Sensitivity – This sets the sensitivity of the optical receiver circuitry
- Maximum Edges – This sets the maximum number of pulses to detect and report
- Detection Mode – This sets the detection mode, which dictates which type of returns to include in the sensor manager point stream. These modes include:
  - First Return Only
  - All Returns
- Power over Ethernet – Power can be applied or turned off for each of the four (4) PoE ports.

*User Note: When setting the inner and outer motor speeds, the inner motor speed must be at least 3 times greater than the outer motor speed in order to satisfy laser safety requirements. Users will be unable to select parameters that do not comply with this requirement.

5.3 3DRi Software Installation

A PC is required for running the 3DRi Viewer and to optionally run Advanced Plugins (when not using a smart OPAL). The 3DRi software runs on Windows 7/10 (64-bit), Ubuntu Linux 14.04 (64-bit), or the embedded Tegra processor in a smart OPAL. Typical and advanced system requirements are shown in

Table 5-1. Advanced systems are recommended for complex machine vision applications (e.g. real-time change detection versus tracking and identifying multiple fast-moving objects concurrently).
### Table 5-1: Typical and Advanced System Requirements

<table>
<thead>
<tr>
<th></th>
<th>Typical System</th>
<th>Advanced System</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operating System</td>
<td>Windows 7/10 (64 bit), Ubuntu Linux 14.04 (64 bit)</td>
<td>Windows 7/10 (64 bit), Ubuntu Linux 14.04 (64 bit)</td>
</tr>
<tr>
<td>Memory</td>
<td>8 GB</td>
<td>16 GB</td>
</tr>
<tr>
<td>Disk Space</td>
<td>300 MB free for program and plugins, 100 GB free for scan data and projects.</td>
<td>300 MB free for program and plugins, 500 GB free for scan data and projects.</td>
</tr>
<tr>
<td>Interfaces</td>
<td>100mbps Ethernet, USB 2.0</td>
<td>Gigabit Ethernet, USB 2.0</td>
</tr>
<tr>
<td>Processor</td>
<td>Intel i3 dual core, 2.13 GHz</td>
<td>Intel i7 quad core, 3.96 GHz</td>
</tr>
</tbody>
</table>

Your purchase of an OPAL Sensor includes a USB flash drive containing the 3DRi software. The drive contains both an .exe installer file for the Windows 7/10 operating system and an install package for the Ubuntu Linux 14.04 operating system.

#### 5.3.1 Installing on Windows 7/10

Insert the USB flash drive into your PC, right-click on the OPAL_3DRi_Software-x.y.z-win64.exe (where x,y,z are version numbers) file and select “Run as Administrator” to launch the program. (If Windows displays a prompt to allow the program to make changes to your computer then select the “Yes” button.) The OPAL 3DRi Software Setup Wizard will then be displayed (Figure 5-5).
Select the “Next” button, and the Software License Agreement will be displayed. Review the license terms, and if you accept them, select the “I Agree” button. The next screen (Figure 5-6) allows you to choose the install location for the 3DRi software.

If you wish to change the default destination folder for the software, select the Browse button and then use the file browser to select the desired directory. Once the Destination Folder is
displayed, select the Next button. The next screen allows you to choose where the software's shortcuts will be displayed in the Start Menu.

Customize the shortcuts for the Start Menu folder if you desire; otherwise use the defaults by selecting the Next button. The following screen allows you to choose the 3DRi components that will be installed.

![Choose Components Screen](image)

**Figure 5-7: The OPAL 3DRi Software Setup Wizard’s Choose Components Screen**

Note that by default, all the components are checked for installation. If you have purchased a license for only the core components, you can still install the higher-level components like Obscurants etc., but without a license those higher-level components will not run when you start the System Manager. Only the components you have purchased will start, and a dialog will report “Feature Not Found” for the other components.

Once you have selected the components' checkboxes, select the “Install” button and the installation process will begin. After a minute or two, a message will indicate that the installation was completed successfully.

### 5.3.2 Installing on Ubuntu Linux 14.04

Connect the USB flash drive to your Ubuntu Linux 14.04 workstation. Browse to the USB drive, and there will be an `OPAL_3DRi_Software-x.y.z-Linux.sh` file there (where x,y,z are version numbers). Copy this file to a directory on the workstation. Open a command shell, and change directories to the location where the install file was copied.

The install file is a shell script and it must be run using Administrator privileges. If you are not logged in as the administrator, but have administrator privileges, use the `sudo` command to run the shell script. (Note also that the script must have execute permissions. Use the command `chmod a+x <filename>` if the copied file does not have execute permissions.)
In the next figure, the file has been copied to the /opt directory, and the sudo command is about to be used to run the install file from the current directory.

![Image of terminal output showing file permissions and sudo command]

**Figure 5-8: Running the Linux install script from a command prompt**

When the install script is run, the Software License Agreement is displayed. Use the space bar or the Enter key to scroll through and read the Agreement. At the end of the Agreement is a prompt asking if you accept the terms of the Agreement. If you accept the terms of the Agreement, select the ‘y’ key. The next prompt will indicate that the software will be installed in a sub-directory of the /opt directory. If you select the ‘n’ key at this prompt, the software will be installed at the top level of the /opt directory. It is recommended that you select the ‘y’ key so that the 3DRi software is installed in its own sub-directory. After choosing whether or not to use a sub-directory, the extraction process will then begin. After a few moments, a message will indicate that the unpacking finished successfully, and a new Linux prompt will be displayed.
Figure 5-9: A message indicates that the unpacking finished successfully

The 3DRi software has now been installed in the specified directory under /opt.

5.3.3 Obtaining a license key

The installation package for the 3DRi Software also includes a utility that will generate a file that can be sent to Neptec to activate your license for that workstation. This utility is called the Remote Update System (RUS), and is a Windows executable. If your workstation is running the Ubuntu Linux 14.04 operating system, the RUS utility must be run under Wine (an application that allows Windows applications to run on Unix-like operating systems). Before running the RUS utility, the Licensing Run-Time Environment must be installed. The following paragraphs describe the installation method for Windows and Ubuntu Linux.

5.3.3.1 Installing the Licensing Run-Time Environment for Windows

Connect the USB flash drive to your PC and open the Licensing Tools folder. Copy the “haspdinst.exe” utility to a local directory. Open a command prompt and change directories to the local directory where the utility has been copied. From the command prompt, issue the following command:

```plaintext
haspdinst.exe -i
```
If Windows displays a prompt asking whether to allow the application to make changes to the computer, select Yes. A dialog will then indicate that the Run-Time Environment is being installed:

![Sentinel Run-time Environment Installer](image)

**Figure 5-10: The Licensing Run-Time Environment is being installed**

After a few moments, the dialog will indicate that the installation is complete:

![Sentinel Run-time Environment Installer](image)

**Figure 5-11: The Licensing Run-Time Environment installation is complete**

Select the OK button, and the dialog will close. The Run-Time Environment has now been installed. Proceed to the next section, “Running the RUS Utility”.

### 5.3.3.2. Installing the Licensing Run-Time Environment for Linux

Connect the USB flash drive to your PC and open the Licensing Tools folder. This folder contains the package for the Run-Time Environment as well as a shell script that installs the 32-bit compatibility package for the Run-Time Environment.

The shell script has the following filename:

```
install_32bit_compatibility_package_for_x64.sh
```

The shell script in the Licensing Tools folder is inside of a tar file of the same name. Copy this tar file to a local directory, and open a terminal window in that local directory. Expand the tar file by typing the following command:

```
tar -xvf install_32bit_compatibility_package_for_x64.tar
```

Expanding the tar file will create a sub-directory that contains the shell script. Change into that sub-directory. Ensure that the script has execute permission by issuing the following command:

```
chmod a+x install_32bit_compatibility_package_for_x64.sh
```

Then, as root, issue the following command:

```
./install_32bit_compatibility_package_for_x64.sh
```
The compatibility package has now been installed. Next, install the package for the Run-Time Environment. The package is in the Licensing Tools folder on the USB flash drive and has the following filename format:

`aksusbd-7.x.x-i386.tar.gz`

Copy this file to a local directory, and open a terminal window in that local directory.

Type the following command to expand the package:

```
tar -zxvf aksusbd-7.x.x-i386.tar.gz
```

(Note that when the command is issued, the letter ‘x’ in the above line must be replaced by the numbers in the actual filename.)

Change directories into the new directory:

```
 cd aksusbd-7.x.x-i386
```

As root, issue the following command:

```
./dinst
```

The Run-Time Environment has now been installed. As a confirmation that the installation took place successfully, verify that the following files are now installed:

- `/usr/sbin/winehasp`
- `/usr/sbin/hasplmd`
- `/etc/init.d/aksusbd`
- `/var/hasplm/haspvlib_xxxxx.so`

where ‘xxxxx’ is a five-digit code.

Additionally, run the following command:

```
sudo /etc/init.d/aksusbd status
```

And verify that the output reports that the required license daemons are running:

- **AKSUSB is running.**
- **WINEHASP is running.**
- **HASPLM is running.**

Having verified that the Run-Time Environment has been installed, proceed to the next section, “Running the RUS Utility”.
5.3.3.3. Running the RUS Utility

Browse to the Licensing Tools folder on the USB flash drive and locate the utility called “RUS.exe”. Double-click on the executable to launch the utility (see Figure 5-12).

![Figure 5-12: The Remote Update System utility](image)

Ensure that the “Collect Status Information” tab is selected, and select the radio button for “Installation of new protection key”.

If this workstation already has a Neptec license key already installed, the “Installation of new protection key” radio button will be greyed out (as shown in Figure 5-12). If that is the case, select the “Update of existing protection key” radio button.

Next, click on the “Collect Information” button. A dialog will be displayed entitled “Save Key Status As”, as shown in Figure 5-13.
Choose a folder (or the Desktop) in which to save the file, and specify a file name (e.g., computer-info). Select the “Save” button, and the data will be written to the file name you specified, with the .c2v extension (for example, computer-info.c2v).

E-mail this .c2v file to your Customer Support representative. You will receive back an e-mail that contains a license file (with the extension .v2c). Save this file to a folder on the same workstation where you generated the .c2v file.

Launch the RUS utility again, and select the “Apply License File” tab (Figure 5-14).

Select the “…” button. A dialog will be displayed entitled “Select the file to apply”, as shown in Figure 5-15.
Figure 5-15: Select the File to Apply dialog

Browse to the folder where the .v2c file is located. Select the file, and click on the “Open” button. The file name will be displayed in the RUS utility’s “Update File” field (Figure 5-16).

Figure 5-16: Apply Update

Select the “Apply Update” button, and after a few moments RUS will indicate that the update has been applied successfully.

The license key is now installed on the workstation, and the 3DRi software can be run.

5.3.4 Upgrading your software

It is recommended that before installing a new updated version of the software, you should uninstall the current version that is on your workstation.
If you have purchased additional features, you will receive a new installer executable. As with an update of the software, it is recommended that you first uninstall the current version of the software, and then install the new software. After installing the new features, it will be necessary to update your license key using the RUS utility, as previously discussed in Section 5.3.3.3. When the RUS utility is run this time, the radio button for “Installation of new protection key” will be greyed out; the radio button for “Update of existing protection key” will be enabled. The other steps for using the RUS utility are the same as in Section 5.3.3.3.

5.4 3DRi Software Architecture

For the OPAL sensor, Neptec provides a 3DRi Software Development Kit (SDK) consisting of:

- the 3DRi System Manager
- optional 3DRi Plug-ins for additional functionality.

3DRi uses an open “publish-subscribe” architecture where the components (“Plug-ins”) communicate with each other through a lightweight Ethernet-based data distribution service. This allows you to build applications that utilize only the Plug-ins that are required for your particular application, and makes it easy to scale the application for deployment on a single computer to a series of networked computers.

The 3DRi SDK is available for Windows 7/10 (64-bit), Ubuntu Linux 14.04 (64-bit), and the embedded Tegra processor.

![Figure 5-17: 3DRi Software Architecture for OPAL Sensors](image)

The System Manager component is the parent process for all of the other sub-processes that are configured to run. Its responsibility is to launch the sub-processes and to monitor their health. If a sub-process terminates unexpectedly, or hangs, the System Manager component will restart it. It is to be run on a separate workstation or the optional embedded Tegra processor).
In addition to the 3DRi System Manager, a graphical user interface (GUI) is available. This GUI is called the 3DRi Viewer. The Viewer can be used to manually interact with the OPAL, allowing you to issue scan commands and view acquired scan data.

### 5.5 3DRi Plug-ins

The 3DRi Software Development Kit supports various optional Plug-ins that provide additional functionality. The Plug-ins are organized into two groups:

- **Core Plug-ins** that provide functionality commonly used in 3D sensor applications
- **Advanced Plug-ins** that can be used by OEMs to develop intelligent real-time applications. They include features such as automatic change detection, object recognition and tracking.

#### Table 5-2: List of Core and Advanced Plug-ins

<table>
<thead>
<tr>
<th>Core Plug-ins</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register</td>
<td>Register your 3D point clouds to external navigation data (GPS and IMU).</td>
</tr>
<tr>
<td>Segment</td>
<td>Segment your registered OPAL 3D data into ground and above-ground data</td>
</tr>
<tr>
<td>Obscurants</td>
<td>Enable the obscurant-penetration capabilities of your OPAL Sensor. This Plug-in detects the presence of obscurants in the raw data, controls the OPAL system’s obscurant mode and removes returns from obscurants from the 3D data.</td>
</tr>
<tr>
<td>Viewer</td>
<td>View your OPAL 3D data. Features include 3D point cloud visualization and manipulation, range/elevation colourization, and basic feature measurements. The Viewer also seamlessly integrates with the System Manager GUI for sensor control and management of the installed Plug-ins.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Advanced Plug-ins</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model</td>
<td>Manage your 3D data from up to four OPAL Sensors in a coherent database</td>
</tr>
<tr>
<td>Align</td>
<td>Automatically align 3D data from different scans in real-time and without reference markers.</td>
</tr>
<tr>
<td>Objects</td>
<td>Segment and classify above-ground objects as stationary or moving objects, or by size.</td>
</tr>
<tr>
<td>Detect</td>
<td>Automatically detect changes in your 3D data in real-time</td>
</tr>
<tr>
<td>Identify</td>
<td>Identify objects using a database of known objects. The Plug-in outputs object ID, type and a confidence factor.</td>
</tr>
<tr>
<td>Track</td>
<td>Track objects in real-time. The Plug-in outputs object ID, position, speed and heading.</td>
</tr>
</tbody>
</table>

### 5.6 3DRi Register Plugin

This section provides an overview of the components that make up the 3DRi Register package and the input and output data as it pertains to configuration of the Core Components. This section also outlines the steps required to configure the 3DRi Register package by working through two separate use cases. The first use case deals with an OPAL Sensor mounted on a moving platform where there is an external INS system providing navigation data and time synchronization. The second use case examines the instance where the OPAL Sensor is in a
static location that has been surveyed so that the latitude, longitude and altitude of the sensor are known.

Figure 5-18 shows the components that make up the 3DRi Register package and the pertinent data inputs and outputs.

![3DRi Register Package](image)

**Figure 5-18: 3DRi Register Package**

5.6.1.1. **Transform Server**

This component performs two functions, as described below.

The first function is to convert navigation information received through the API into the local frame 3DRi navigation data format, which is published on a dedicated channel so that it can be used by other 3DRi components which use navigation data. The API navigation data format data contains Yaw, Pitch and Roll information as opposed to the quaternion rotation data used by the components.

The second function is to publish the static transforms between different frames of reference, which are read from a configuration file at start-up; e.g. the transform between a vehicle frame and the sensor frame. It also contains a static location/orientation solution which can be manually set in cases when dynamic global location data is not available. The static orientation solution will only be applied if the GPS position is set to something other than the origin (0,0,0) and if no other GPS position is provided elsewhere. The transformation data is published for use by other components.

5.6.1.2. **Register**

This component performs the transformation of points from a local frame to a geo-registered coordinate frame. Register receives Internal Local Frame Navigation data, Static Transform data, and Cartesian Sensor data as input streams. It uses the input data streams to transform the Cartesian Sensor points into Geo-registered points which it publishes on its output channel listed in
Table 5-3. Register uses the timestamp attributes in both the Cartesian Sensor Points and the Internal Local Frame NAV data in order to synchronize and accurately register the data.

<table>
<thead>
<tr>
<th>Publish Channel</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OPAL2_Register_Points</td>
<td>Registered Points</td>
</tr>
</tbody>
</table>

5.6.2 OPAL Sensor Coordinate Conventions

Figure 5-19 shows the orientation of the OPAL coordinate axes. The positive Z axis is out the sensor lens (and out of the top of the panoramic OPAL dome), the positive X axis points down (into the page) and the positive Y axis points to the left. The origin of the sensor frame (0,0,0) for the OPAL is the center of the crosshair located on the top surface of the sensor (see red circle in Figure 5-19).

![Figure 5-19: OPAL Coordinate System](image)

5.6.3 User Navigation App Requirements

In this scenario, it is required that in addition to the 3DRi software there is also an application that is converting the navigation data output by the INS system into the format required by the 3DRi API.

The User Nav App should be created using the 3DRi API. This is a program which converts from some external INS format to the format which the Transform Server can use. For additional details in writing software that uses the API refer to Section 7.0.
The program should do the following actions using the API:

- Fill in an ExternalNavInput_t data structure and call the function C3DRIClientControl::updateNav(ExternalNavInput_t). This publishes the message on the OPAL2_3DRiClient_Applanix_Nav channel for Transform Server to use.

- Create a GPZDA string representing the GPS/INS time and call the function C3DRIClientControl::SetTime(GPZDA). This sets the time on the sensor to match the GPS/INS time.

- Attach the PPS output of the INS system to the Sensor. This will closely synchronize the time between the sensor and the INS system. Without this synchronization the accuracy will decrease and performance will be reduced.

5.6.4 Configure 3DRi Software for Moving Platform

The following steps outline the configuration changes that are required to the SystemManager.xml file to configure all of the 3DRi components. Before starting this configuration step make sure that the 3DRi System Manager has been shut down.

The following list summarizes the changes to the default configuration that need to be made to enable the 3DRi Software to support scanning on a moving platform:

1. Enable execution of Register
2. Enable execution of Transform Server
3. Configure Transform Server to receive its NAV data
4. Define the Transformation from the sensor’s frame of reference to the navigation system frame of reference
5. (Optional) Set any additional static Transformations required.

Step 5 is optional; it needs only to be performed if you require additional static transformations. To perform this step the additional transforms have to be added by using a text editor and opening the SystemManager.xml file and copying and pasting the transform data blocks marked by the <Transform_1> </Transform_1> tags. Once copied the start and end tags need to be updated to Transform_2, Transform_3 et cetera depending on how many are added. Once they have been added it is recommended that the file be opened again with the ConfigEditor to make the final changes to ensure that the completed file is compatible with the 3DRi software.

To complete the steps listed above you need to run the ConfigEditor application and load the SystemManager.xml. In Figure 5-20, the field that requires change to complete Step 1 is highlighted. The Run Register field gets set to Yes.
In Figure 5-21 the field to complete step 2 is highlighted. The Run Transform Server field needs to be set to Yes.

In steps 3 and 4 the subscription channel to the navigation data is defined and the transformation from the sensor frame to the navigation system frame is defined. This step will be illustrated through a simple example as follows.
Figure 5-22: Illustration showing relationship of SensorFrame0 to NavFrame1

Figure 5-22 shows a hypothetical relationship between an OPAL sensor and the reference frame of the Navigation system where there is a displacement of 10m in the Y axis and a positive rotation of 90 degrees about the Z axis (using a right hand rule rotation quaternion).

In Figure 5-23 the fields to complete steps 3 and 4 are highlighted. The input Navigation Subscribe channel must be updated to OPAL2_NavData_INS_ExternalNavConverter. Transform_1’s SourceFrame is set to SensorFrame0 and its TargetFrame is set to NavFrame1. This single transform is the difference between the frame of the sensor (SensorFrame0: X forward, Y left, Z up, optical origin) and the frame of the provided navigation pose (NavFrame1). TransformServer provides pose in the WorldFrame.

Figure 5-23: Steps to Configure OPAL on a Moving Platform
Once all of the configuration steps have been completed you need to save the SystemManager.xml file and exit the ConfigEditor. The system is now ready to run. To operate the OPAL sensor, start the 3DRi System Manager. It will initiate all of the processes that have been configured. The user will need to then start the External NAV App that will feed Navigation data to the 3DRi software through the API.

A typical display of geo-referenced data is shown in the viewer in Figure 5-24. The XYZ axis is shown in the top left of the screen. The X-axis is East, the Y-axis is North and the Z axis is up. Clicking on a point on the screen shows the geographic coordinates on the top left. The sensor is shown in its calculated orientation and position.

![Figure 5-24: Display of Geo-Referenced Data](image)

### 5.6.5 OPAL at a Known Fixed GPS Location

The section outlines the configuration changes to operate an OPAL sensor at a fixed location with a known GPS location to generate geo-referenced point cloud data. In this scenario there is no requirement for a User NAV App to run.

Before starting this configuration step, make sure that the 3DRi System Manager has been shut down.

The following list summarizes the changes to the default configuration that need to be enabled so that the 3DRi Software can support scanning on a moving platform:

1. Enable execution of Register
2. Enable execution of Transform Server
3. Set fixed position GPS position of the Sensor
4. Define the transformation from the sensor’s frame of reference to the world frame of reference

To complete the steps listed above you need to run the ConfigEditor application and load the SystemManager.xml. In Figure 5-25 the field that requires change to complete Step 1 is highlighted. The Run Register field gets set to Yes.

![Figure 5-25: Step 1 Configure OPAL sensor for a Fixed Position](image)

In Figure 5-26 the fields that require changes to complete Steps 2 and 3 are highlighted. The Run Transform Server field gets set to Yes, the UseSpecifiedOrigin field gets set to Yes, and the Latitude, Longitude and Altitude are set to the surveyed position of the OPAL Sensor. Latitude and Longitude units are in decimal degrees and Altitude is in units of meters.
In Figure 5-27 the fields to complete step 4 are highlighted. Transform_1’s SourceFrame is set to SensorFrame0 and its TargetFrame is set to WorldFrame. This single transform is the difference between the frame of the sensor (SensorFrame0: X forward, Y left, Z up, optical origin) and the frame of the specified origin (WorldFrame). TransformServer provides pose in the WorldFrame. If additional static transforms are required between the specified origin and the sensor orientation then they need to be set under the ReferenceFrames fields. The rotational components are set as unit quaternions.
Once all of the configuration steps have been completed you need to save the SensorManager.xml file and exit the ConfigEditor. The system is now ready to run. To operate the OPAL sensor, start the 3DRi System Manager and it will initiate all of the processes that have been setup.

5.7 3DRi Obscurants Plugins

This section provides an overview of the components that make up the 3DRi Obscurants package and the input and output data as it pertains to configuration of the Core Components.

Figure 5-28 shows the components that make up the 3DRi Obscurant package and the pertinent data inputs and outputs.
5.7.1.1. 3DRi Ground Filter

The Ground Filter component is the first component of the 3DRi Obscurants package. It is designed as a first step in segmenting data for analysis. It can receive either Cartesian Sensor Points or Geo-registered Points as an input; it then outputs the data as either “Above-Ground” or “Ground Points”. The data outputs can be used with other 3DRi modules or accessed through the API. The filtered points from this module must be used as the inputs for the 3DRi SDF application.

<table>
<thead>
<tr>
<th>Publish Channel</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OPAL2_GF_Ground_Points</td>
<td>Points filtered to be part of the ground plane</td>
</tr>
<tr>
<td>OPAL2_GF_Above_Ground_Points</td>
<td>All other points not part of ground plane</td>
</tr>
</tbody>
</table>

5.7.1.2. Scan Data Filtering (SDF)

The Scan Data Filtering (SDF) component is the next component in the 3DRi Obscurants package. SDF can operate on either Cartesian Sensor Points or Geo-registered Points. The selection of which input is used is based on the configuration file. The configuration file is loaded at start up and the filtering process begins when points begin to arrive on the input channels. Once processing is up and running, the SDF will filter the points to remove noise and perform a basic level of data segmentation. The output of SDF is a set of points that have been classified as object objects, and a set of points that the filter has removed as being noise. Depending on the selection of the configuration file, SDF can also control the OPAL Scanner by issuing scan update commands to switch between Clear and Dust modes when the noise condition encountered warrants it.
5.7.2 Configuring Obscurants

This section describes the steps required to configure the 3DRi Obscurants package. In order to properly configure 3DRi Obscurants you need to decide how the OPAL sensor will be gathering data, and how obscured the conditions will be under the worst case during scanning.

Delivered with the 3DRi Software, there is a directory named “3DRi Obscurant Config Files”. Use the descriptions in Table 5-6 to decide which configuration best describes the anticipated scanning conditions. The selected configuration file needs to be copied to the folder where the 3DRi Software was installed and then renamed to SDF.conf.

<table>
<thead>
<tr>
<th>Table 5-6: 3DRi Obscurant Configurations</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Configuration File Name</strong></td>
</tr>
<tr>
<td>SDF_moving.conf</td>
</tr>
<tr>
<td>SDF_moving_dust.conf</td>
</tr>
<tr>
<td>SDF_static.conf</td>
</tr>
<tr>
<td>SDF_static_dust.conf</td>
</tr>
</tbody>
</table>
### Configuration File Name | Description
--- | ---
Register processing being performed. The sensor is in a static position and its GPS location is not known.

This configuration performs heavy filtering and is for use when worst case conditions result in moderate to heavy obscuration. This configuration has the option enabled for 3DRi Obscurants to automatically switch the sensor to the most appropriate detection mode based on the conditions.

Once the configuration file has been copied and renamed in the 3DRi Software folder, the SystemManager.xml file needs to be updated to enable 3DRi Obscurant processing.

Before starting this configuration step make sure that the 3DRi System Manager has been shut down.

Run the ConfigEditor application and load the SystemManager.xml. Change the Run SDF field and Run GroundFilter field to Yes as shown in Figure 5-29.

![Figure 5-29: Configure 3DRi Obscurants to Run](image)

Once the configuration step has been completed you need to save the SystemManager.xml file and exit the ConfigEditor. The system is now ready to run. To operate the OPAL sensor, start the 3DRi System Manager and it will initiate all of the processes that have been setup.
6.0 Operating the OPAL Sensor

In this section, it is assumed that the OPAL Sensor has been correctly installed and that the 3DRi System Manager software has been loaded onto the sensor and correctly configured.

The OPAL can be controlled through either the Application Programing Interface (API) or the 3DRi Viewer (a graphical user interface). Please see Section 7.0 API User Guide, for details on using the API to issue scan commands. This following procedure provides an overview of how to use the 3DRi Viewer to issue scan commands and view the data.

6.1 Start the OPAL Sensor

Apply power to the sensor. The sensor will perform its power-up sequence, including the laser safety checks, which will take one to two minutes. Once the sensor has started its power-up sequence, proceed to the next section.

6.2 Start 3DRi SystemManager (optional)

If there are any 3DRi modules to be run with the OPAL hardware, make sure they are configured following the module instructions in Section 5.5, then launch SystemManager from the OPAL 3DRi Software folder.

6.3 Start the 3DRi Viewer

Select the shortcut to the 3DRi Viewer under the OPAL 3DRi software folder on the start menu. Launch this application to display the 3DRi Viewer (Figure 6-1).

Figure 6-1: The 3DRi Viewer
6.4 Select a Scan Pattern

Four pre-set scan patterns are available that can be selected in order to start acquiring data without configuring any scan parameters. (Figure 6-2). To access these scan patterns, select the Basic button in the Sensor Control Panel. These scan patterns are called:

- Lower Speed Short Range
- Lower Speed Long Range
- Max Speed Short Range
- Max Speed Long Range

In addition you can select the Advanced button to allow you to configure the individual scan parameters. To return to the pre-set dropdown selection, press the Basic button.

![Figure 6-2: Sensor Control Panel Showing pre-set scan options](image)

6.5 Customizing the Scan Parameters

The scan parameters can be customized to specify the length of a scan, the speed at which the motors move, the laser power, and more. The parameters can be modified in the Sensor Control Panel (Figure 6-3) of the 3DRi Viewer.
Operating the OPAL Sensor

Figure 6-3: The Sensor Control Panel

Details on each of the Scan Parameters are as follows:

- **Scan Duration**
  - The number of seconds for which scanning is to take place. For example, if the scan duration is set to 30 (valid range is from 0.0 to 86400.0 seconds) and the Start Scan button is selected, the sensor will acquire data for 30 seconds. Setting the value to 0 would initiate a continuous scan when the Start Scan button is selected. The sensor will remain operational and continue to acquire data until the Stop Scan button is selected.

- **Inner Motor RPS**
  - The number of revolutions per second made by the Inner Motor.

- **Outer Motor RPS**
  - The number of revolutions per second made by the Outer Motor.

- **Laser Power**
  - The strength of the emitted laser, expressed as a percentage of the maximum laser power. Setting this value to 100 will cause the laser to be emitted at maximum power.

- **Scan Rate**
  - The number of laser pulses emitted each second. This parameter is set by using a drop-down menu. The options are 300 kHz (or 300,000 points per second), 200 kHz, 100 kHz. This setting also affects the maximum range. In 300 kHz mode the maximum detection range is \(~450\)m, at 200 kHz the maximum detection range is \(~750\)m, and at 100 kHz the maximum detection range is \(~1000\)m.

- **Detection Mode**
  - This sets the detection mode, which dictates which type of returns to include in the sensor manager point stream. These modes include:
Operating the OPAL Sensor

- First Return Only
- First and Last Returns
- All Returns

- Detection Sensitivity
  - This sets the sensitivity of the optical receiver circuitry

- Maximum Edges
  - This sets the maximum number of pulses to detect and report.

6.6 Performing a Scan

In the 3DRi Viewer, select the “Start Scan” button. The OPAL sensor will take a few moments to spin its motors up to the required speed, after which scan data will begin to be added to the 3DRi Viewer display.

Note that when scanning is underway, the name of the “Start Scan” button changes to “Update Scan”. This new button provides the ability to change the scan parameters without having to stop and start the scan operation again.

![Sensor Control: 1003](image)

**Figure 6-4: The Update Scan button is available during scanning**

If the scan duration was set to a non-zero number (i.e. a fixed duration), scanning will stop when the number of seconds has elapsed. If the Scan Duration was set to zero, scanning will continue until commanded to stop. To stop scanning at any time, select the Stop Scan button.

6.7 Using the Workspace Panel

When scanning is started, a new Workspace item is added to the list in the Workspace Panel.
As shown in Figure 6-5, the Workspace name is the timestamp for when the data began to be acquired; in this case, on Tuesday September 22, 2015 at 13:16:34.

A single Workspace item represents all of the data acquired during a single scan. If a scan is stopped and a new scan is started, a new Workspace item is added to the list in the Workspace Panel.

The checkbox beside each Workspace item indicates whether or not the data in that Workspace is displayed. By default, all Workspace items are displayed unless you deselect a checkbox.

### 6.8 Saving the Scan Data

If you have performed several scans, each scan will be listed as a separate dataset in the Workspace panel. To save all of the scans to a series of PIF files, select the File drop-down menu and select the Save As item. A dialog is then displayed to specify the folder to which the files will be written, as well as the name of the xml file that will list each of the PIF files that are saved.

To save just one of the scans to a PIF file, right-click on the dataset name in the Workspace panel, select Export from the menu that is displayed, and then select “To Pif” from the sub-menu.

To save one of the scans to an ASCII file, right-click on the dataset name in the Workspace panel, select Export from the menu that is displayed, and then select “To ASCII” from the sub-menu. Selecting this option will export the Cartesian (xyz) point data to a comma-separated file.

To save one of the scans to a PCD file, right-click on the dataset name in the Workspace panel, select Export from the menu that is displayed, and then select “To Pcd” from the sub-menu. Selecting this option will export the Cartesian (xyz) point data to a PCD or Point Cloud Data format, which is compatible with the Open Source Point Cloud Library (PCL) format.
Multiple datasets can also be exported to a single file. To use this capability, make sure that the checkboxes for all the datasets that you wish to export are selected and confirm that the checkboxes for any datasets that you do not wish to export are deselected. Then, right-click on one of the selected datasets, select Export Checked from the menu that is displayed, and then select one of “To ASCII”, “To Pif” or “To Pcd” from the sub-menu. A new file will be created, in the format you have selected, that combines the points from all of the selected datasets into one single file.

6.9 Displaying the Scan Data

As scan data is acquired, it is added to the display in the centre of the 3DRi Viewer. You can change the way in which the data is shown, as detailed in the following sections. All of the following actions can be performed while scanning is underway, or after scanning has stopped.

6.10 Using the Mouse

6.10.1 Zoom, Rotate and Shift the Data

The mouse can be used to change the view of the data as follows:

- **Zoom**
  - To change the degree of zoom, hold down the right button on the mouse and drag. Moving the mouse downwards zooms out on the data and moving the mouse upwards zooms in on the data. The degree of zoom can also be changed by moving the mouse scroll wheel up or down.

- **Zoom in on a specified area**
  - To zoom in on a specified area, hold the Shift button and then hold down the middle mouse button on the mouse and drag to create the specified area.

- **Rotate**
  - To rotate the data, hold down the left button on the mouse and drag. Moving the mouse downwards rotates the data towards the user and moving the mouse upwards rotates the data away from the user. Moving the mouse left or right rotates the data left or right.

- **Shift**
  - To shift (or translate) the data, hold down the middle button on the mouse and drag. Moving the mouse in a given direction shifts the data in the same direction.

- **Spin**
  - To spin the data about the current view axis, hold the Shift button and then hold down the left button on the mouse and drag.

- **Snip from Area**
  - To create a new sub-set point data set, hold down the Ctrl button and hold down the middle mouse button and drag a box over the points of interest.
When the middle mouse button is released the points that were selected will create a new point data set that will be selectable in the Workspace Panel.

6.10.2 Perform a Measurement Between Two Points

To measure the distance between any two points in the displayed scan data, hold down the Control button, then click with the left mouse button on the first point and drag the mouse to the second point. As long as the left mouse button is held down, the mouse can continue to be moved to other points, and the displayed distance will continue to be updated. The displayed distance is presented in metres.

6.11 Displaying the List of 3D Controls

The 3DRi Viewer provides a list of the controls. This list can be accessed by selecting the Help drop-down menu and selecting the Controls item:

![Figure 6-6: The Help drop-down menu](image)

Selecting the Controls item will display the Controls dialog:

![Figure 6-7: The Controls dialog](image)
6.12 Displaying the Reference Frame Axes

The reference frame axes can be displayed by selecting the Display drop-down menu and then selecting the Reference Frame item.

![Displaying the Reference Frame](image)

**Figure 6-8: Displaying the Reference Frame**

If this item is selected, the x, y and z axes will be shown on the scan data display area, providing an easy visual cue to the orientation of the data. To remove the reference frame axes from the scan data display area, select the Display drop-down menu and then select the Reference Frame item again.

6.13 Viewing the Data along a Specified Coordinate Axis

The scan data can be viewed along a specified coordinate axis, by right-clicking on the active dataset name in the Workspace panel, selecting the View item and then choosing the desired view.

![Specifying the Coordinate Axis](image)

**Figure 6-9: Specifying the Coordinate Axis**

Note that the convention used for selecting the specified axis view is that a positive axis is pointed towards the viewer. For example, selecting “+x” will orientate the reference frame and the scan data with the positive x-axis pointing towards the viewer.
6.14 Viewing the Data within a Specified Area of Interest

The scan data that is displayed (and added to the current dataset) can be limited to a specified area of interest in the field of view. This functionality can be useful if the operator wishes to obtain a high density of points around a certain object or area without acquiring a massive amount of data for the sensor’s entire field of view.

In the example in the next Figure, scan data is being acquired showing a parking lot with trees beyond the lot.

![Scan data acquisition](image)

Figure 6-10: Scan data is being acquired of an entire parking lot

To create a polygon that defines an area of interest, right-click on the active dataset and select Modify/Polygon Segmentation as shown in Figure 6-11: Selecting Polygon Segmentation from the Modify menu. (Note that this functionality can only be used on a dataset that is still active. An older dataset from a previous scan cannot have an area of interest specified.)
Figure 6-11: Selecting Polygon Segmentation from the Modify menu

Having selected Polygon Segmentation, the polygon tool becomes active, and the cursor can be used to define a polygon for the area of interest, as shown in the next Figure. The points of the polygon are created by left-clicking the mouse on the points in the scan display.

Figure 6-12: Creating a polygon on the scan display

When the polygon is “closed” by clicking the last point over the first point, the polygon turns green, and all scan data outside of that polygon is removed from the display and the active
dataset. Figure 6-13: Only scan data within the polygon is kept shows scan data being received only within the specified polygon.

**Figure 6-13: Only scan data within the polygon is kept**

Note that the lines that make up the polygon extend in the positive and negative directions along the z-axis, creating the sides of the three-dimensional area of interest. Any scan data that falls within that area of interest is kept. Figure 6-14: A top-view of the polygon defining the area of interest shows a top view of the acquired scan data, with the positive z-axis pointing towards the viewer.
6.15 Viewing the Scan Data in Radar Plot Mode

The scan data that is displayed (and added to the current dataset) can be viewed in a Radar Plot mode in which the data’s reference frame is oriented with the z-axis vertical. An overlay shows the angles that comprise the field of view and the range from the sensor.

In the example in Figure 6-15: Selecting the Radar Plot item from the Display drop-down menu, scan data is being acquired showing a mining site. The user has selected the Display drop-down menu and is choosing the Radar Plot item.
After the Radar Plot item is selected, a dialog is shown that provides the ability to configure the display, as shown in Figure 6-16: The Radar Plot Parameters dialog.

The options for the FOV are in units of degrees, and the user can select from 360°, 120°, 90° or 60° degrees. The Angle option provides the ability to choose the increment size of the angle elements of the overlay: the user can choose between having an increment at every 5 degrees or every 15 degrees. The next two parameters configure the rings of the overlay that indicate the range from the origin. The Range Increment Size defines the distance, in metres, that each ring of the overlay represents. The # of Concentric Rings defines the total number of rings that will be shown on the overlay. Finally, there is a checkbox for Remove Angle Values. If this checkbox is left in its default state of unchecked, the overlay will show the values (e.g. from 0 to
Operating the OPAL Sensor

360) that correspond to each angle element; if the checkbox is selected, the values will not be shown.

After configuring the parameters, select the OK button on the dialog and the scan data will then be displayed in Radar Plot mode, as shown Figure 6-17: The scan data is presented in Radar Plot mode.

![Figure 6-17: The scan data is presented in Radar Plot mode](image)

Note that when Radar Plot mode is enabled, Track Movement is enabled as well. If the sensor is providing Navigation Data, the Radar Plot will update the position of the sensor and the scan display’s orientation as new Navigation Data is received.

6.16 Using the Display Properties Panel

The Display Properties Panel offers the ability to configure how the scan data is displayed to the user.

6.16.1 The Apply Button

At the bottom of the Display Properties Panel (Figure 6-18) is the “Apply” button. After making any changes to a display property dataset, select the “Apply” button. Note that any changes that have been made will be discarded if the user switches between datasets without selecting the “Apply” button.
6.16.2 Selecting a Dataset

The first property on the list is “Dataset”, and in the example, its value is “Default”. The properties assigned to the Default dataset will be used on any new scans that are taken after the changes have been applied. To make changes to the display properties of scans that have already been taken, click with the left-mouse button in the Workspace Panel on the dataset of interest. The Display Properties Panel will then update with the properties for that scan, and the Dataset field will now show the name of that scan:

![Image of Display Properties Panel]

Figure 6-18: The Apply Button
Figure 6-19: An Updated Display Properties Panel

Note that the final item in the above Figure 6-19, # Points, specifies how many points of data have been acquired for that dataset. The user cannot change this number. If the Default set of display properties is selected, the # Points will read “N/A”.

To switch back to the Default set of display properties, click with the left-mouse button in the Workspace Panel on the name of the sensor (above the list of datasets). The Display Properties Panel will then update with the default properties, and the Dataset field will once again read Default.

6.16.3 Selecting the Type

The second item on the Display Properties Panel is Type (Figure 6-20), and this field allows the user to choose to display the data according to range, elevation, or intensity, or to use a single colour for the data. To select a type, double-click on the current value of the Type and a drop-down menu will be displayed:
6.16.3.1. Colour by Range

If the Type is set to Colour by Range (Figure 6-21), the colour (or brightness) of each data point on the display will be based on the horizontal distance of that data point from the sensor.

Other properties that can be selected will depend on the Type that has been chosen. Any of these other properties can be selected by double-clicking on the current value. For Colour by Range, the user can choose:

- Colourization – either grey scale (with black as the Minimum and white as the Maximum) or Coloured (with red as the Minimum and light blue as the Maximum)
- Minimum and Maximum range – in meters
- Outside Range – determines whether data points outside of the Minimum and Maximum values will be Hidden or Shown
- Live View – determines whether only the most recently acquired data points will be displayed. If the user selects Off, then all data acquired during the scan will be displayed. If the user selects Time Limited, an additional property field allows the user to select how many seconds of recently acquired data to display. If the user selects Point Limited, an additional property field allows the user to select how many points of recently acquired data to display.
- Point Size – a scaling factor that allows the adjustment of the size of the individual points of scan data on the display. If the Point Size is set to 5, the individual points of scan data will be five times as large as they were with the Point Size set to 1.
Note that choosing to have Live View as On will have an effect only if it is chosen for the Default dataset (for any subsequent scans that are taken) or for the current dataset that still has scan data being acquired. For any dataset that has finished scanning, all of the scan data will continue to be displayed.

6.16.3.2. Colour by Intensity

If the Type is set to Colour by Intensity (Figure 6-22), the colour (or brightness) of each data point on the display will be based on the intensity of the laser returned by that data point.

For Colour by Intensity, the user can choose the following other properties:

- Colourization – either Grey scale (with black as the Minimum and white as the Maximum) or Coloured (with red as the Minimum and light blue as the Maximum)
- Minimum and Maximum Intensity – between 0 and 255
- Live View – determines whether only the most recently acquired data points will be displayed. If the user selects Off, then all data acquired during the scan will be displayed. If the user selects Time Limited, an additional property field allows the user to select how many seconds of recently acquired data to display. If the user selects Point Limited, an additional property field allows the user to select how many points of recently acquired data to display.
- Point Size – a scaling factor that allows the adjustment of the size of the individual points of scan data on the display. If the Point Size is set to 5, the individual points of scan data will be five times as large as they were with the Point Size set to 1.
### 6.16.3.3. Colour by Elevation

If the Type is set to Colour by Elevation (Figure 6-23), the colour (or brightness) of each data point on the display will be based on the vertical distance of that data point from the sensor.

For Colour by Elevation, the user can choose the following other properties:

- **Colourization** – either greyscale (with black as the Minimum and white as the Maximum) or Coloured (with red as the Minimum and light blue as the Maximum)
- **Minimum and Maximum elevation** – in meters
- **Outside Range** – determines whether data points outside of the Minimum and Maximum values will be Hidden or Shown
- **Live View** – determines whether only the most recently acquired data points will be displayed. If the user selects Off, then all data acquired during the scan will be displayed. If the user selects Time Limited, an additional property field allows the user to select how many seconds of recently acquired data to display. If the user selects Point Limited, an additional property field allows the user to select how many points of recently acquired data to display.
- **Point Size** – a scaling factor that allows the adjustment of the size of the individual points of scan data on the display. If the Point Size is set to 5, the individual points of scan data will be five times as large as they were with the Point Size set to 1.

![Display Properties Table](image)

**Figure 6-22: Colour by Intensity settings**

<table>
<thead>
<tr>
<th>Properties</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dataset</td>
<td>Default</td>
</tr>
<tr>
<td>Type</td>
<td>Color By Intensity</td>
</tr>
<tr>
<td>Colorization</td>
<td>Colored</td>
</tr>
<tr>
<td>Live View</td>
<td>Off</td>
</tr>
<tr>
<td>Min</td>
<td>0</td>
</tr>
<tr>
<td>Max</td>
<td>235</td>
</tr>
<tr>
<td>Point Size</td>
<td>1</td>
</tr>
<tr>
<td># Points</td>
<td>N/A</td>
</tr>
</tbody>
</table>

Apply
6.16.3.4. Colour Constant

If the Type is set to Colour Constant (Figure 6-24), all data points on the display will use a single colour, to be selected by the user.

For Colour Constant, the user can choose the following other properties:

- **Colour** – the single colour to be applied to all the data points. The user can double-click on the colour shown, and a drop-down menu of available colours is displayed, as shown in Figure 6-24.

- **Live View** – determines whether only the most recently acquired data points will be displayed. If the user selects Off, then all data acquired during the scan will be displayed. If the user selects Time Limited, an additional property field allows the user to select how many seconds of recently acquired data to display. If the user selects Point Limited, an additional property field allows the user to select how many points of recently acquired data to display.

- **Point Size** – a scaling factor that allows the adjustment of the size of the individual points of scan data on the display. If the Point Size is set to 5, the individual points of scan data will be five times as large as they were with the Point Size set to 1.
Operating the OPAL Sensor

6.16.3.5. Colour by Range/Intensity

If the Type is set to Colour by Range/Intensity (Figure 6-25), the colour (or brightness) of each data point on the display will be based on the horizontal distance of that data point from the sensor, and that colour is then scaled based on the data point’s intensity. For example, if the range of a point determines that its colour should be red, but the point’s intensity is very low, the displayed colour for that point will be a dark red. If a different data point at the same range has a very high intensity, the displayed colour for that point will be a bright red.

Other properties that can be selected will depend on the Type that has been chosen. Any of these other properties can be selected by double-clicking on the current value. For Colour by Range/Intensity, the user can choose:

- Colourization – either greyscale (with black as the Minimum and white as the Maximum) or Coloured (with red as the Minimum and light blue as the Maximum)
- Minimum and Maximum range – in meters
- Outside Range – determines whether data points outside of the Minimum and Maximum values will be Hidden or Shown
- Live View – determines whether only the most recently acquired data points will be displayed. If the user selects Off, then all data acquired during the scan will be displayed. If the user selects Time Limited, an additional property field allows the user to select how many seconds of recently acquired data to display. If the user selects Point Limited, an additional property field allows the user to select how many points of recently acquired data to display.
- **Point Size** – a scaling factor that allows the adjustment of the size of the individual points of scan data on the display. If the Point Size is set to 5, the individual points of scan data will be five times as large as they were with the Point Size set to 1.

![Display Properties](image)

**Figure 6-25: Colour by Range/Intensity settings**

### 6.16.3.6. Colour Random

If the Type is set to Colour Random (Figure 6-26), the selected dataset will be coloured a random colour selected by the viewer. If this is the Default setting then each new dataset created will be coloured a new random colour.

![Display Properties](image)

**Figure 6-26: Colour Random**
6.17 Setting the Background Colour

The default Background Colour of the scan display is black, but this colour can be configured using the Background Colour dialog. To access this dialog, select the Display drop-down menu at the top of the 3Dri Viewer (Figure 6-27) and then select the Background Colour item.

![Accessing the Background Colour dialog](image)

Figure 6-27: Accessing the Background Colour dialog

The Background Colour dialog will then be shown (Figure 6-28).

![The Background Colour dialog](image)

Figure 6-28: The Background Colour dialog

Any of the basic colours on the left side of the dialog can be selected by clicking on the colour with the mouse. Similarly, any colour in the spectrum on the right side of the dialog can be selected by clicking on the desired point in the spectrum. These custom colours can also be
Operating the OPAL Sensor

adjusted by changing the numbers for Red, Green and Blue, or Hue, Saturation and Value. (Value represents a greyscale value between 255 for Black and 0 for White.)

A custom colour can also be saved by selecting the Add to Custom Colours button. In Figure 6-29, two custom colours have been saved.

![Image of custom colour selection dialog]

Figure 6-29: Saving a Custom Colour

Once the desired colour has been chosen, select the OK button. The dialog will close and the background colour of the scan display will be set to the selected colour.

6.18 Checking the Log Panel

The Log Panel contains any system messages, warnings or errors reported by the software. In the example below, a series of system messages (Info) has been reported during start-up.

![Image of Log Panel]

Figure 6-30: The Log Panel shows a series of System Messages
If an error or warning is reported, then that tab will be highlighted to indicate that a new error has been reported. In the example below, the Warnings tab has been highlighted, indicating that a new warning has been reported on that tab.

![Image](image.png)

**Figure 6-31: The Warnings tab is highlighted**

The messages, warnings and errors can be cleared from the Log Panel in two ways. The first method is to right-click inside an individual tab, and then to select Clear from the menu that is shown:

![Image](image.png)

**Figure 6-32: Clearing the messages from the Warnings tab**

This method will clear the messages within only that tab. In the example above, the Warnings tab will be cleared, but any messages on the Info tab or Error tab will not be cleared.

The second method for clearing messages from the Log Panel is to select the Display drop-down menu at the top of the 3DRi Viewer and then select the Clear Log item:
Operating the OPAL Sensor

6.19 Using the Point Cloud Registration Tool

The 3DRi Viewer provides a Point Cloud Registration Tool that allows the user to “register” one set of scan data into the same frame of reference as a second set of data. This tool can be used when two datasets are collected when the sensor is at different positions and orientations, and as a result the two datasets are not aligned with each other. The example Figure 6-34: Two datasets that are not aligned with each other illustrates this problem: in the figure, the ground plane for the first set of data (in blue) is almost at a right angle to the ground plane for the second set of data (in red).

Figure 6-33: Clearing the messages from all tabs in the Log Panel

This method will clear the messages from all the tabs. The Info tab, the Warnings tab and the Error tab will all be cleared.

Figure 6-34: Two datasets that are not aligned with each other
To use the tool, select the Point Cloud Registration button at the upper-left corner of the Viewer, as shown in the following Figure. Selecting this button will open the Point Cloud Registration panel.

![Figure 6-35: The Point Cloud Registration button](image)

The registration of the first dataset into the frame of reference of the second dataset is done using a set of points, defined by the user, on common features that exist in both datasets. Note that the points that are chosen should not be in a line in three-dimensional space, as this would make it difficult for the algorithm to converge on the correct alignment of the points.

The Figure 6-36: A top-view of the first dataset shows a top-view of the first dataset. The walls of a building can be seen, and near the top of the view a truck is parked beside one of the walls.

![Figure 6-36: A top-view of the first dataset](image)

To select the set of points for the first dataset, deselect the checkbox for the second dataset. (This is done so that the Point Cloud Registration tool associates the selected points with the first dataset.) Then, zoom in and rotate the data until the identified feature can be easily seen, and then click on the point using the keyboard combination of <Ctrl+Shift> while using the left mouse button.
In Figure 6-37: The first point in the first dataset has been selected a point on the upper-right corner of the truck has been selected. In the Point Cloud Registration panel, a set of x, y and z coordinates has been filled in for the first point in the first dataset.

![Figure 6-37: The first point in the first dataset has been selected](image)

In Figure 6-38: The second point in the first dataset has been selected, the user has shifted the view to another part of the scan area and selected a point on the lower-left corner of a different truck in the parking lot. In the Point Cloud Registration panel, a set of x, y and z coordinates has been filled in for the second point in the first dataset.

![Figure 6-38: The second point in the first dataset has been selected](image)

In Figure 6-39: The third point in the first dataset has been selected, the user has shifted the view to another part of the scan area and selected a point on the lower corner of the building.
seen previously in the top-view of the data. In the Point Cloud Registration panel, a set of x, y and z coordinates has been filled in for the second point in the first dataset.

Additional points can be selected, but a minimum of three is required.

Having selected the three points in the above Figures for the first dataset, the user then disables the checkbox for the first dataset and enables the checkbox for the second dataset. Having done this, the next points to be selected will be associated with the second dataset.

In Figure 6-40: The first point in the second dataset has been selected, a point on the upper-right corner of the first truck has been selected on the second dataset. In the Point Cloud Registration panel, a set of x, y and z coordinates has been filled in for the first point in the second dataset. (Note that the points on the common features in the second dataset must be selected in the same order as in the first dataset.)
Figure 6-40: The first point in the second dataset has been selected

In Figure 6-41: The second point in the second dataset has been selected, the user has shifted the view to another part of the scan area and selected a point on the lower-left corner of the other truck in the parking lot. In the Point Cloud Registration panel, a set of x, y and z coordinates has been filled in for the second point in the second dataset.

Figure 6-41: The second point in the second dataset has been selected

In Figure 6-42: The third point in the second dataset has been selected, the user has shifted the view to another part of the scan area and selected a point on the lower corner of the building seen previously in the top-view of the data. In the Point Cloud Registration panel, a set of x, y and z coordinates has been filled in for the third point in the second dataset.
Having selected three points on common features in both datasets, the Calculate button can now be selected. Before selecting Calculate make sure that both datasets have been enabled for display by selecting the checkmarks beside each in the Workspace Panel.

Figure 6-43: The ICP Input Parameters dialog shows the ICP input parameters dialog that is displayed when the Calculate button is selected. This dialog contains a set of recommended, default parameters for the algorithm.

When the OK button is selected, the algorithm will begin to run. This process can take up to a minute, and while it is underway a message at the lower-left corner of the Viewer will indicate “Running ICP”. When the algorithm finishes running, a new dialog will be displayed showing the Transformation Matrix, Quaternion Rotation and Fitness Score that were calculated for the transformation, as shown in Figure 6-44: The Transformation Results dialog. (These values are also reported in the shell that launched the 3DRi Viewer.)
Selecting the OK button closes the above dialog.

Having completed its run, the algorithm has generated two new datasets in the Workspace Panel, as shown in the next Figure. Both datasets contain the scan data points from the original first dataset, translated and rotated into the reference frame of the second dataset. The names of the two new datasets begin with the prefixes “Fine_” and “Coarse_” and are followed by the name of the original first dataset. The Coarse dataset contains the results after a single iteration of the algorithm and the Fine dataset contains the results after the Maximum Number of Iterations of the algorithm.

As can be seen in Figure 6-45: Fine dataset points have been aligned with the original second dataset, the user has left the checkboxes for only two datasets selected: the original second dataset (with points coloured in pink) and the calculated Fine dataset (with points coloured in yellow).
To exit out of the tool, select the Point Cloud Registration button at the upper-left corner of the Viewer a second time. This will close the Point Cloud Registration panel.

### 6.20 Using the Control Point Registration Tool

The 3DRi Viewer provides a Control Point Cloud Tool that can calculate the transformation between the sensor’s coordinate system and the global coordinate system. This tool can be used when the sensor is not connected to a Navigation source providing a GPS position for the sensor. To use the tool it is necessary for the user to know the GPS position of multiple physical objects in the sensor’s field of view; these positions are the “truth data” that the tool uses to calculate the transformation.

The example in the next Figure provides an example set of scan data that was collected by pointing an OPAL from the rooftop of a building: the field of view shows trees, cars, signposts and other buildings. The cursor has been used to click on a point of data, and the top right of the scan display shows the Cartesian XYZ data for that point; however, the top left of the scan display is blank, showing that no Latitude, Longitude or Elevation information is available.
Figure 6-46: Scan data that contains only Cartesian information

To use the tool, select the Control Point Registration button at the upper-left corner of the Viewer, as shown in Figure 6-47: The Control Point Registration button (It is the right button).

Figure 6-47: The Control Point Registration button

Selecting this button will open the Control Point Registration panel, as shown in Figure 6-48: The Control Point Registration panel.
The registration of the sensor’s coordinate system into the global coordinate system is done using a set of points in the scan data, selected by the user, for which the GPS (Latitude, Longitude and Elevation) is known. Note that the points that are chosen should not be in a line in three-dimensional space, as this would make it difficult for the algorithm to converge on the correct transformation for the sensor.

To select the set of points for the dataset, first ensure that the checkboxes for any other datasets are deselected. (This is done so that the Control Point Registration tool selects the points from the desired dataset.) Then, zoom in and rotate the data until the identified feature can be easily seen, and then click on the point using the keyboard combination of <Ctrl>+<Shift> while using the left mouse button.

Figure 6-49: The first point in the dataset has been selected shows the Control Point Registration panel after the first point has been selected. A set of x, y and z coordinates has been filled in for the first point in the dataset.
The GPS information for the points is entered into fields under the heading of Control Points. This can be done as each individual point is selected or after all the points have been selected. In the next Figure, the Latitude, Longitude and Elevation have been filled in for the first point in the dataset.

**Figure 6-50: GPS information has been defined for the first point in the dataset**

Continue selecting points in the dataset and entering their known GPS information into their associated Control Points field. A minimum of three points is required, but additional points can help the algorithm to converge on the best possible transformation.

If the GPS information for the sensor’s position is known, this too can be entered into the Control Point Registration panel, in the field that contains the greyed-out text “(Optional) Define Origin”. If the GPS information for the sensor’s position is not known, leave this field blank; the algorithm will then use the GPS information for the first point in the dataset as the origin point instead.

In Figure 6-51: GPS information has been defined for five points and the sensor origin, five points of data have been selected and their corresponding GPS information has been entered, as has the GPS information for the sensor’s origin.
Operating the OPAL Sensor

Figure 6-51: GPS information has been defined for five points and the sensor origin

If an incorrect point was selected, it can be removed by clicking in that row of the Control Point Registration panel and selecting the Delete Selected button.

Once the GPS information for all points and (optionally) the sensor has been entered, the tool will calculate the transformation when the user selects the Calculate button. When this button is selected, a dialog will be displayed showing the results of the calculation, as presented in the Figure 6-52: A dialog displays the calculated transformation

Figure 6-52: A dialog displays the calculated transformation

Note that the Source Frame will match the frame of the SensorManager points dataset that has been used; in this case, it was SensorFrame0. If multiple sensors were being used, and the dataset that was used originated from SensorFrame2, the calculated transformation would be from SensorFrame2.

As shown in Figure 6-52: A dialog displays the calculated transformation, the tool has calculated both the Translation and the Quaternion rotation that defines the transformation between the sensor’s Source Frame and the global Target Frame. This transformation will be exported to the configuration file TransformServer.xml (if you wish to export to a different file, simply modify the displayed filename). To export to the configuration file, select the Create Config button.
The 3DRi folder will now contain a TransformServer.xml file with the calculated transformation. To make use of this transformation, copy the transformation information from the exported configuration file into the TransformServer section of SystemManager.xml.

Once Register and TransformServer are running with their configurations correctly defined, use the GUI to add a new Point Source called OPAL2_Register_Points. The new point source will contain scan data with both Cartesian and GPS information.

6.21 Using the SystemLogger Panel

The 3DRi Software includes a SystemLogger component that writes all data to a file on the local computer's hard drive. SystemManager's configuration file is set by default to launch SystemLogger on startup, and by default the log file is written to the same folder as the 3DRi Software. The name of the log file begins with the prefix “System_log_” and is followed by the date on which the log file is first created. If SystemLogger is shut down and then restarted, the application will look in the 3DRi folder for an existing log file and, if it exists, open it and begin appending new data to it.

All of the parameters that configure SystemLogger can also be modified through the 3DRi Viewer. If the GUI's SystemLogger panel is not already displayed, select the View drop-down menu and select SystemLogger. The panel will then be displayed. Figure 6-53: The SystemLogger panel shows the SystemLogger panel with its default values.

![System Logger Panel](image)

The parameters that can be modified are as follows:

- **Log_On_Startup** - a Boolean flag that can be set to Yes or No. It defines whether or not SystemLogger will open and write to a file when the application is started.

- **Channels** - a list of the data channels to be logged. The default value for this parameter is ".*", which means that all channels will be logged. If you wish to log only, for example, OPAL2_Register_Points, enter that channel name in this field and no other channels will be logged.
• Exclude_Channels - a list of the data channels that are not to be logged. By default this parameter is left blank, which means that no channels will be excluded. If you wish to exclude, for example, OPAL2_SensorManager_Points0, enter that channel name in this field and then that one channel will not be logged.

• Log_Directory - the name of the sub-directory inside of the 3DRi folder where SystemLogger should write its log file. Note that SystemLogger will not create this sub-directory on its own; the user must manually create it before SystemLogger is restarted with this parameter changed.

• Log_Filename - the prefix to be used for the name of each log file that is created.

• Log_Duration - the number of minutes for which logging should run. The default value for this parameter is 0, which means that logging will run indefinitely.

• Log_Size_Limit - the maximum allowable size of a single log file measured in MB. If the current log file is about to reach this size, SystemLogger will close that file and open a new one. The default value for this parameter is 1000 MB.

• Total_Logging_Size_Limit - the maximum allowable size of all log files in the current logging directory. If the total size of all the log files is about to reach this size, SystemLogger will delete the oldest log file that it can find. The default value for this parameter is 4000 MB.

Note that when any of these parameters are modified on the SystemLogger panel, it is necessary to stop and then start logging again in order for SystemLogger to use the updated parameters. So, make the desired changes to the parameters in the SystemLogger panel, and then select the Stop Logging button. The button will change to read Start Logging. Select that button, and SystemLogger will start logging again, this time using the new parameters.

The log files written by SystemLogger are in a binary format that can be played back using the LcmLogPlayer utility included in the API zip file found on the install media. Note that if all channels have been logged and those channels are played back, any Start Scan or Stop Scan commands that were logged will be issued again, so it is important not to have SensorManager running when the log file is played back, otherwise these scan commands will be sent to the OPAL sensor again.
7.0 3DRi Client Application Programming Interface (API)

The 3DRi Client API package can be found on the 3DRi Software installation disk, or can be downloaded from the customer portal on the Neptec website. http://neptectechnologies.com/

7.1 Operating Requirements

- The 3DRi Client API has been tested with the following configurations:
  - On Windows 7/10 with Visual Studio 2013 64-bit version
  - On Ubuntu 14.04 Linux with gcc 4.8.x 64-bit version
- CMake
  - The example program included with the API is configured to use CMake to generate native makefiles in Linux or Visual Studio workspaces in Windows. CMake version 3.0.0 or higher is required. CMake can be downloaded from http://www.cmake.org/.
- UDP Multicast Setup
  - You must have a multicast-enabled network interface and a valid multicast route must be defined. See https://lcm-proj.github.io/index.html for more information.

7.2 Contents of Installation Disk API Folders

- API Folder
  - This folder contains CMake scripts used for generating the native build environment for compiling the example program.
- ClientExample Folder
  - This folder contains an example program that can be built and run in either Linux or Windows. The program demonstrates some basic functionality of the API.
- Vendor Folder
  - The Vendor folder contains the shared libraries required to build a program that uses the API.
- API_Doxygen Folder
  - Contains API reference documentation in HTML format.
- Sim Folder
  - Contains a Windows-only application that can be used to simulate the OPAL2 sensor for testing purposes. Also contains a copy of the SensorManager and PointCalibrator application that provides a connection between the sensor and programs that are communicating with the sensor.
7.3 Building the Example Program

The example program that is built by completing the instructions that follow, demonstrates how the API can be used to connect to an OPAL sensor, to start the sensor scanning and to save the received points to an ASCII comma-separated file.

7.3.1 For Windows

1. Launch cmake-gui – see Figure 7-1
2. Set the source location to the API folder (where the top-level CMakeLists.txt file is located).
3. Choose a build location for the binary output.

4. Click “Configure” button.
5. CMake may ask to create the build directory if it doesn’t exist. If so, click “Yes”
6. Specify Visual Studio 12 2013 Win64 as the generator for the project and use default native compilers (Figure 7-2) and click on the “Finish” button.

Figure 7-2: Specify generator and compiler (64-Bit Version)

7. To link with debug libraries, change “CMAKE_BUILD_TYPE” from “Release” to “Debug” (see Figure 7-3). Click “Configure” a second time to accept all of the default values shown in red.
Building the Example Program

Figure 7-3: Link to Debug Libraries

8. Click “Generate”

9. Run the Soln_3DRI_API_EXAMPLE_<Configuration>.bat script in the build location for the binary output (where Configuration is either Debug or Release). This will load the 3DRI_API_EXAMPLE.sln solution file, generated by CMake, into Visual Studio and set the PATH environment variable so that the required library DLLs will be located when running the example.

10. **Important** – In Visual Studio, check that the Solution Configuration (Release or Debug) is set to match the CMAKE_BUILD_TYPE specified. This will ensure that the build links to the appropriate libraries.

11. In Visual Studio, right click on the ClientExample project and choose “Set as StartUp Project”.

12. In Visual Studio, build the solution.
7.3.2 For Linux

1. Launch cmake-gui – see Figure 7-4
2. Set the source location to the API folder (where the top-level CMakeLists.txt file is located).
3. Choose a build location for the binary output.

![Figure 7-4: Configuring CMake-gui in Unix](image)

4. Click “Configure” button.
5. CMake may ask to create the build directory if it doesn’t exist. If so, click “Yes”
6. Specify Unix Makefiles as the generator for the project and use default native compilers.
7. To link with debug libraries, change “CMAKE_BUILD_TYPE” from “Release” to “Debug” (see Figure 7-6). Click “Configure” a second time to accept all of the default values shown in red.
8. Click “Generate”.
9. Open a terminal and change to the directory where the Makefile was generated:

   $ cd /home/opal2/NTC_API_BuildExample

10. Build the example:

    $ make

### 7.4 Running the Example Program

#### 7.4.1 On Windows Using the Simulator

The simulator is a Windows-only application that simulates the OPAL panoramic sensor.
1. Before starting the SensorManager.xml file needs to be updated to change the IP address of the sensor to 127.0.0.1.

2. For running the example program on Windows, launch SensorManager, PointCalibrator and the Simulator using the “Run Simulator.bat” script provided in the Sim folder. This will start both the simulator and the SensorManager and PointCalibrator programs.

3. Run the client example on the Windows computer. This can be done either within Visual Studio or from a command prompt. There are batch files in the ClientExample build directory for opening up a command prompt such that the PATH environment variable is set to locate the DLLs used by the example program. For release, use Cmd_ClientExample_Release.bat and for debug use Cmd_ClientExample_Debug.bat

7.4.2 On Linux Using the Simulator

1. For running the example program on Linux, you will need a Windows computer to run the simulator. Launch only the simulator in Windows by running Scanner360Simulator.exe.

2. On the Linux computer, update the <IP_Address> parameter under <OPAL2_Connect_Parameters> in the SensorManager.xml file in the Sim folder with the IP address of the Windows computer running the simulator.

3. Open a terminal and run SensorManager on the Linux computer
   $ cd NTC_API_Delivery/Sim
   ./SensorManager

4. Open another terminal and run the ClientExample program
   $ cd NTC_API_BuildRelease/ClientExample
   $ ./ClientExample
7.4.3 Tips

Only one instance of SensorManager and PointCalibrator should be running at any given time.

Changes to the XML configuration file are not automatically detected. If changes are made to the SensorManager.xml configuration, close SensorManager and PointCalibrator and restart them so that the new configuration will be loaded.

If you are building and running the example program on Linux or running the simulator on a computer different to the one that is running the example program, update the TTL parameter in the XML file from 0 to 1 to ensure the UDP traffic will transmit across the local network.

In Linux, if the following warning appears

** (process:NNNNN): WARNING **: LCM UDP receive buffer size (262142)
is smaller than requested (20000000). For more info:

https://lcm-proj.github.io/multicast_setup.html

change the UDP receive buffer size with the following commands:

$ sudo sysctl -w net.core.rmem_max=20000000
$ sudo sysctl -w net.core.rmem_default=20000000

or change permanently by adding the following lines in /etc/sysctl.conf:

net.core.rmem_max=20000000
net.core.rmem_default=20000000

7.5 Technical Overview of the API

The 3DRi Client (API) is a set of C++ classes designed to allow access to OPAL 3DRi functionality. All classes and data structures provided through the API are defined under the OPAL2::Client3DRi namespace.

For details on a specific class or API function, please refer to the Doxygen-generated HTML help files.

The API makes use of the following open source libraries:

- Boost, for threading and shared pointers
  - Version 1.56
  - [http://www.boost.org/users/history/version_1_56_0.html](http://www.boost.org/users/history/version_1_56_0.html)
- Lightweight Communications and Marshaling (LCM), for underlying communications between processes
  - Version 1.1.2
Building the Example Program

- https://github.com/lcm-proj/lcm/releases

- Zlib for data compression
  - Version 1.2.8
  - http://www.zlib.net/

- Geographic lib for latitude, longitude, altitude conversions
  - Version 1.37

7.5.1 Class Diagram

Figure 7-8 illustrates the most common classes used to interface with the 3DRi system. For a complete listing of all available classes and data types, please refer to the Doxygen HTML documentation.
Figure 7-8: Most Common OPAL3 Classes to Interface with 3DRI system
7.5.1 Workflow

Figure 7-9 illustrates the general workflow of the example program included with the API. It demonstrates the end to end life cycle of an application that uses the API to gather point data from a sensor.

![Workflow Diagram]

**Figure 7-9: Workflow Diagram**

7.5.2 Initialization

7.5.2.1 Control Object

The control object C3DRiClientControl is a singleton class that provides access to other objects in the API. A reference to the one and only control object can be obtained using C3DRiClientControl::GetInstance().

The client application must initialize the Control object using C3DRiClientControl::Initialize() before it can be used. In addition, before the client application exits, C3DRiClientControl::Cleanup() must be called to shut down the API functionality.
Following initialization, C3DRiClientControl::PopulateDatasources() should be called until it returns success. This gives the system time to detect and enumerate all of the data sources that are available. The 3DRi software must be running in order for data sources to be detected.

### 7.5.3 Point Data Source

C3DRiClientControl::GetDataSources() can be called to obtain a vector containing a list of all of the available data sources. The list consists of C3DRiClientDataSource objects which are generic base-class. The objects in the list can be cast to their true types by querying C3DRiClientDataSource::GetType().

The following code fragment shows how to obtain an OPAL60 data source pointer from the list of data sources:

```cpp
using namespace OPAL2::Client3DRi;

C3DRiClientControl& control = C3DRiClientControl::GetInstance();
.
.
.
// Request a vector that contains shared pointers to all of the available data sources.
DataSourcePtrVector vectDataSources;
err = control.GetDataSources(vectDataSources);
if (ResultSuccess != err)
{
    std::cout << "Unable to obtain list of data sources [" << err << "]" << std::endl;
    return -1;
}

// Find the OPAL2 data source and cast data source pointer to use it
OPAL2Ptr pOPAL2;
DataSourcePtrVector::iterator it = vectDataSources.begin();
while (it != vectDataSources.end())
{
    if (it->getType() == OPAL2_SENSOR)
    {
        pOPAL2 = std::static_pointer_cast<C3DRiClientOPAL2,
                                         C3DRiClientDataSource>(*it);
        if (pOPAL2->GetID() == nSensorID)
        {
            break;
        }
        pOPAL2.reset();
    }
    ++it;
}
```
// Verify that the cast worked and that we have a valid pointer
if (pOPAL2.get() == NULL)
{
    std::cout << "Unable to detect OPAL2 sensor with ID = "
              << nSensorID << std::endl;
    return -1;
}

7.5.3.1. OPAL Data Source
The C3DRiClientOPAL3 objects can be used to start and stop scanning and to access the scan parameters for the OPAL3 sensors.

7.5.3.2. OPAL3 Status
An observer that satisfies the IOPAL3StatusObserver interface can be registered using C3DRiClientOPAL3::RegisterStatusObserver() to receive notification whenever the status of the OPAL3 sensor changes. Only one observer can be registered at a time.

IOPAL3StatusObserver::StatusUpdate() gets called when the status changes. Note that StatusUpdate() is an asynchronous call and is made from a separate thread. StatusUpdate() provides a pointer to a COPAL3Status object which is derived from C3DRiClientProperties. C3DRiClientProperties objects provide generic methods to set and get data values using std::string data.

The list of OPAL3 status fields are shown in the table below.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Range</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>State</td>
<td>PROP_LIST</td>
<td>Unknown, Powered Off, Initialized, Standby, Scanning, Failure, Preparing To Scan, Recoverable Error</td>
<td>Current STATE of the Sensor</td>
</tr>
<tr>
<td>Acquiring</td>
<td>PROP_BOOL</td>
<td>Yes, No</td>
<td>Flag to indicate if the Sensor is actively acquiring points. Can be No when STATE is SCANNING during state transition.</td>
</tr>
<tr>
<td>Sensor Type</td>
<td>PROP_LIST</td>
<td>OPAL120, OPAL360</td>
<td>Scanner Type</td>
</tr>
<tr>
<td>Variable</td>
<td>Type</td>
<td>Value</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------</td>
<td>---------------------</td>
<td>----------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Laser_Active</td>
<td>PROP_BOOL</td>
<td>Yes, No</td>
<td>Flag to indicate that active communication exists with the sub-system.</td>
</tr>
<tr>
<td>Laser_Status</td>
<td>PROPINTEGER</td>
<td>0-65535</td>
<td>Status bits provided to indicate operating status of the laser sub-system.</td>
</tr>
<tr>
<td>Laser_Supply_Voltage</td>
<td>PROPFLOAT_POINT</td>
<td>Variable</td>
<td>Supply voltage in volts as measured by the laser sub-system.</td>
</tr>
<tr>
<td>Laser_Temperature</td>
<td>PROPFLOAT_POINT</td>
<td>Variable</td>
<td>Temperature in °C as measured by the laser sub-system.</td>
</tr>
<tr>
<td>Laser_Optical_Power</td>
<td>PROPFLOAT_POINT</td>
<td>Variable</td>
<td>Laser output power in mW.</td>
</tr>
<tr>
<td>Laser_PRF</td>
<td>PROPINTEGER</td>
<td>25000-300000</td>
<td>The pulse repetition frequency at which the laser is currently operating.</td>
</tr>
<tr>
<td>Motor_Active</td>
<td>PROP_BOOL</td>
<td>Yes, No</td>
<td>Flag to indicate that active communication exists with the sub-system.</td>
</tr>
<tr>
<td>Motor_Bus_Voltage</td>
<td>PROPFLOAT_POINT</td>
<td>Variable</td>
<td>Supply voltage in volts as measured by the motor sub-system.</td>
</tr>
<tr>
<td>Motor_Bus_Current</td>
<td>PROPFLOAT_POINT</td>
<td>Variable</td>
<td>Supply current in amps as measured by the motor sub-system.</td>
</tr>
<tr>
<td>Motor_Bus_Power</td>
<td>PROPFLOAT_POINT</td>
<td>Variable</td>
<td>Power in watts being consumed by the motor sub-system.</td>
</tr>
<tr>
<td>Motor_PCB_Temperature</td>
<td>PROPFLOAT_POINT</td>
<td>Variable</td>
<td>Temperature in °C as measured on the motor PCB.</td>
</tr>
<tr>
<td>Motor_Laser_MCU</td>
<td>PROPINTEGER</td>
<td>0-65535</td>
<td>Laser safety bits as determined by the motor microcontroller. A value of 0 means safe for laser to emit.</td>
</tr>
<tr>
<td>Motor_Laser_PLD</td>
<td>PROPINTEGER</td>
<td>0-65535</td>
<td>Laser safety bits as determined by the motor programmable logic. A value of 0 means safe for laser to emit.</td>
</tr>
<tr>
<td>Motor_Inner_State</td>
<td>PROPINTEGER</td>
<td>0-65535</td>
<td>Field indicating the state of the inner motor.</td>
</tr>
<tr>
<td>Motor_Inner_Status</td>
<td>PROPINTEGER</td>
<td>0-65535</td>
<td>Field indicating the status of the inner motor.</td>
</tr>
<tr>
<td>Motor_Inner_Encoder</td>
<td>PROPINTEGER</td>
<td>0-65535</td>
<td>Encoder value of the inner motor.</td>
</tr>
<tr>
<td>Motor_Inner_Temperature</td>
<td>PROPFLOAT_POINT</td>
<td>Variable</td>
<td>Temperature in °C as measured on the motor PCB.</td>
</tr>
<tr>
<td>Variable</td>
<td>Type</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>--------------------------</td>
<td>---------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>Inner_Motor</td>
<td>PROP_FLOAT</td>
<td>Velocity of the Inner Motor measured in RPS</td>
<td></td>
</tr>
<tr>
<td>Motor_Outer_State</td>
<td>PROP_INTEGRER</td>
<td>Field indicating the state of the outer motor.</td>
<td></td>
</tr>
<tr>
<td>Motor_Outer_Status</td>
<td>PROP_INTEGRER</td>
<td>Field indicating the status of the outer motor.</td>
<td></td>
</tr>
<tr>
<td>Motor_Outer_Encoder</td>
<td>PROP_INTEGRER</td>
<td>Encoder value of the outer motor.</td>
<td></td>
</tr>
<tr>
<td>Motor_Outer_Temperature</td>
<td>PROP_FLOAT</td>
<td>Temperature in °C as measured on the motor PCB.</td>
<td></td>
</tr>
<tr>
<td>Outer_Motor</td>
<td>PROP_FLOAT</td>
<td>Velocity of the Outer Motor measured in RPS</td>
<td></td>
</tr>
<tr>
<td>PSM_Active</td>
<td>PROP_BOOL</td>
<td>Flag to indicate that active communication exists with the sub-system.</td>
<td></td>
</tr>
<tr>
<td>PSM_Input_Voltage</td>
<td>PROP_FLOAT</td>
<td>Supply voltage in volts as measured by the power supply sub-system.</td>
<td></td>
</tr>
<tr>
<td>PSM_Input_Current</td>
<td>PROP_FLOAT</td>
<td>Supply current in amps as measured by the power supply sub-system.</td>
<td></td>
</tr>
<tr>
<td>PSM_52V_Output</td>
<td>PROP_FLOAT</td>
<td>Output voltage in volts as measured by the power supply sub-system on the 52V rail.</td>
<td></td>
</tr>
<tr>
<td>PSM_48V_Output</td>
<td>PROP_FLOAT</td>
<td>Output voltage in volts as measured by the power supply sub-system on the 48V rail.</td>
<td></td>
</tr>
<tr>
<td>PSM_24V_Output</td>
<td>PROP_FLOAT</td>
<td>Output voltage in volts as measured by the power supply sub-system on the 24V rail.</td>
<td></td>
</tr>
<tr>
<td>PSM_12V_Output</td>
<td>PROP_FLOAT</td>
<td>Output voltage in volts as measured by the power supply sub-system on the 12V rail.</td>
<td></td>
</tr>
<tr>
<td>PSM_Temperature</td>
<td>PROP_FLOAT</td>
<td>Temperature in °C as measured by the power supply sub-system</td>
<td></td>
</tr>
<tr>
<td>RxM_Active</td>
<td>PROP_BOOL</td>
<td>Flag to indicate that active communication exists with the sub-system.</td>
<td></td>
</tr>
<tr>
<td>RxM_Status</td>
<td>PROP_INTEGRER</td>
<td>Status bits provided to indicate operating status of the receiver sub-system.</td>
<td></td>
</tr>
<tr>
<td>RxM_Mode</td>
<td>PROP_INTEGRER</td>
<td>Currently set sensitivity mode: 0 – Medium Sensitivity</td>
<td></td>
</tr>
</tbody>
</table>
7.5.3.3. Scanning

The OPAL sensor can be commanded to start scanning using the C3DRiClientOPAL3::StartScan() method. If the scan duration is finite, C3DRiClientOPAL3::WaitForScan() can be called and will block until the scan completes or an error occurs. For an infinite length scan, C3DRiClientOPAL3::StopScan() can be called to stop the scan. A scan duration of zero denotes an infinite length scan.
7.5.4 Parameters

Parameter classes are also derived from the C3DRiClientProperties base class. The vector containing all available parameter names can be queried using C3DRiClientProperties::GetAllPropertyNames(). Property names can then be used to determine the property data types and to query and modify individual property values.

For a complete list of C3DRiClientProperties methods, refer to the Doxygen documentation.

7.5.4.1. OPAL3 Scan Parameters

The OPAL3 scan parameters control the behavior of the OPAL3 sensor while scanning.

A copy of the OPAL3 parameters can be obtained by calling C3DRiClientOPAL2::GetParameters() and the parameters on the OPAL3 can be updated by calling C3DRiClientOPAL2::SetParameters().

The available OPAL3 parameters are shown in Table 7-2.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Range</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scan_Duration</td>
<td>PROP_FLOATING_POINT</td>
<td>0.0 – 86400.0 seconds</td>
<td>Length of time to collect data (0.0 for unlimited)</td>
</tr>
<tr>
<td>Inner_Motor_RPS</td>
<td>PROP_FLOATING_POINT</td>
<td>6.667 – 80 rotations/sec</td>
<td>Rotation speed of the inner motor</td>
</tr>
<tr>
<td>Outer_Motor_RPS</td>
<td>PROP_FLOATING_POINT</td>
<td>1.667 – 30.0 rotations/sec</td>
<td>Rotation speed of the outer motor</td>
</tr>
<tr>
<td>Laser_Power</td>
<td>PROP_FLOATING_POINT</td>
<td>0.0 – 100.0</td>
<td>Percentage of the full power to output</td>
</tr>
<tr>
<td>Scan_Rate</td>
<td>PROP_LIST</td>
<td>25 kHz, 50 kHz, 100 kHz, 200 kHz, 300 kHz</td>
<td>Number of points acquired per second.</td>
</tr>
<tr>
<td>Detection_Mode</td>
<td>PROP_LIST</td>
<td>First Pulse Only, First and Last Pulses, Last Pulse Only, All Pulses</td>
<td>Modes to select how much data is returned from the sensor.</td>
</tr>
<tr>
<td>Detector_Sensitivity</td>
<td>PROP_LIST</td>
<td>Low, Medium, High, Ultra</td>
<td>Setting for the sensitivity for the receiver in the OPAL3. Medium should be used for most applications; to maximize ranging High mode is used. Ultra may result in more unwanted noise points as sensitivity is maximized. Neither High nor Ultra should be used.</td>
</tr>
</tbody>
</table>
7.5.5 Receiving Points from a Data Source

C3DRiClientControl::CreatePointDataSource() can be used to create an object that receives point data on a named channel. Once a C3DRiClientPointSource object is created, observers that satisfy the I3DRiClientPointDataObserver interface can be registered to receive notification when points are available.

Multiple observers can be registered for a point source and I3DRiClientPointDataObserver::DataAvailable() is called for each registered observer whenever data new data from the specified subscription channel has arrived.

Note that DataAvailable() is an asynchronous call and is made from a separate thread.

7.5.5.1. Point Data

Cartesian point data from the OPAL sensor arrives on the "OPAL2_SensorManager_Points" channel. These are essentially raw points in the sensor's frame of reference.

The following code fragment is an example of a point data observer derived from I3DRiClientPointDataObserver:

```c++
#include <stdio.h>
#include <3DRiClient/Include/3DRiClientPointDataObserver.h>

/// Class used to monitor the arrival of data from the sensor
class CMyDataObserver : public OPAL2::Client3DRi::I3DRiClientPointDataObserver
{
public:
    CMyDataObserver();
    virtual ~CMyDataObserver();

    /// Observer callback
    virtual void DataAvailable(const OPAL2::Client3DRi::PointDataPtr& data);
};

void CMyDataObserver::DataAvailable(const OPAL2::Client3DRi::PointDataPtr& data)
{
    OPAL2::Client3DRi::PointsVector vecPts;
}```
Building the Example Program

if (OPAL2::Client3DRi::ResultSuccess == data.get()->GetPoints(vecPts, true))
{
    for (OPAL2::Client3DRi::PointsVector::iterator it = vecPts.begin();
        it != vecPts.end(); ++it)
    {
        // Check for invalid points
        if ((*it).X != 0.0 && (*it).Y != 0.0 && (*it).Z != 0.0)
        {
            // Convert time from microseconds to seconds
            double dTime = static_cast<double>((*it).timestamp) / 1000000.0;
            ::printf("%.8f,%.8f,%.8f,%i,%.6f\n",
                static_cast<unsigned int>((*it).intensity), dTime);
        }
    }
}

The following is an example of registering the CMyDataObserver object, as defined above, to receive points on the "OPAL2_SensorManager_Points" channel:

using namespace OPAL2::Client3DRi;

PointSourcePtr pPointSource;
err = control.CreatePointDataSource(pPointSource, "OPAL2_SensorManager_Points");
if (ResultSuccess != err)
{
    std::cout << "Unable to create point source [" << err << "]" << std::endl;
    return -1;
}

boost::shared_ptr<CMyDataObserver> pChannelObserver(new CMyDataObserver);
pPointSource->RegisterDataObserver(
    boost::static_pointer_cast<I3DRiClientPointDataObserver>(pChannelObserver));
### Blob Point Data

In applications that use the 3DRi Objects plug-in it is possible to subscribe to another point data type. The blob point data provides both point data and additional object information to support the tracking of above ground objects that have been segmented by 3DRi Objects.

The following code fragment is an example of a blob point data observer derived from i3DRiClientBlobPointDataObserver.

```c++
#include <stdio.h>
#include <string>
#include <3DRiClient/Include/3DRiClientTypes.h>
#include <3DRiClient/Include/3DRiClientBlobPointDataObserver.h>
#include <boost/thread/recursive_mutex.hpp>

/// Class used to monitor the arrival of data from the sensor
class CMyBlobDataObserver : public OPAL2::Client3DRi::I3DRiClientBlobPointDataObserver
{
  public:
    CMyBlobDataObserver(OPAL2::Client3DRi::SensorIDType sensorID);
    virtual ~CMyBlobDataObserver();

    /// Observer callback
    virtual void DataAvailable(const OPAL2::Client3DRi::BlobPointDataPtr& data);

    void SetGlobalFrame(bool globalFrame) { m_bGlobalFrame = globalFrame;};

  void CMyBlobDataObserver::DataAvailable(const OPAL2::Client3DRi::BlobPointDataPtr& data)
  {
    OPAL2::Client3DRi::PointInfo_t info;
    OPAL2::Client3DRi::BlobPointInfo_t blobInfo;

    if ((OPAL2::Client3DRi::ResultSuccess == data->GetPointInfo(info))&&
        (OPAL2::Client3DRi::ResultSuccess == data->GetBlobPointInfo(blobInfo)))
    {
      // Filter out points on this channel that aren't from the desired sensor
      if (m_sensorID == info.sensorID)
      {
        OPAL2::Client3DRi::BlobPointsVector vecPts;
        if (OPAL2::Client3DRi::ResultSuccess == data.get()->GetPoints(vecPts, m_bGlobalFrame))
        {
          m_uiTotalPoints += vecPts.size();

          std::stringstream ss;
          ss << "Added " << std::setw(6) << vecPts.size() << " pts, Total = " <<
          std::setw(8) << m_uiTotalPoints << " to " << m_strFilename << std::endl;
          std::cout << ss.str();

          boost::recursive_mutex::scoped_lock lock(m_mutex);
          if (m_pFP != NULL)
          {
            ::fprintf(m_pFP, "%s,%i,%.4f,%.4f,%.4f,%.4f,%.4f,%.4f\n",
```
for (OPAL2::Client3DRi::BlobPointsVector::iterator 
     it = vecPts.begin();
     it != vecPts.end();
     ++it)
{
    if ((*it).X != 0.0 && (*it).Y != 0.0 && (*it).Z != 0.0)
    {
        // Convert time from microseconds to seconds
        double dTime = static_cast<double>(*it).timestamp) / 
                       1000000.0
        ::fprintf(m_pFP,
               "%.8f,%.8f,%.8f,%i,%.6f
               \n",
               (*it).X,
               (*it).Y,
               (*it).Z,
               static_cast<unsigned int>(*it).intensity),
               dTime);
    }
}

7.5.5.3. Other Point Sources

Other point data sources may be available depending on the system configuration. Table 7-3 shows a list of some other point data sources that can be observed:

<table>
<thead>
<tr>
<th>Process</th>
<th>Channel</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3DRi Register</td>
<td>OPAL2_Register_Points</td>
<td>Geo-referenced points based on the Navigation data.</td>
</tr>
<tr>
<td>3DRi Obscurants</td>
<td>OPAL2_GF_Ground_Points</td>
<td>Geo-referenced segmented ground points.</td>
</tr>
<tr>
<td>3DRi Obscurants</td>
<td>OPAL2_GF_Above_Ground_Points</td>
<td>Geo-referenced segmented above ground points.</td>
</tr>
<tr>
<td>Process</td>
<td>Channel</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>----------------------------</td>
<td>------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>3DRi Obscurants</td>
<td>OPAL2_SDF_Filtered_Points</td>
<td>Geo-referenced points that the 3DRi Obscurants plug-in has identified as noise and have been removed.</td>
</tr>
<tr>
<td>3DRi Obscurants</td>
<td>OPAL2_SDF_Points</td>
<td>Geo-referenced segmented above ground object points.</td>
</tr>
<tr>
<td>3DRi Objects</td>
<td>OPAL2_Blob_Points</td>
<td>Geo-referenced above ground segmented objects with additional data to manage object tracking.</td>
</tr>
</tbody>
</table>

### 7.5.5.4. Coordinate System

`I3DRiClientPointDataObserver::DataAvailable()` provides data as a pointer to a `C3DRiClientPointData` object. The data points are copied from this object to a local vector using `C3DRiClientPointData::GetData()`. `GetData()` includes an additional parameter to specify whether the points should be in a local East-North-Up coordinate frame with a local origin or converted to global Latitude, Longitude and Altitude. Transform information, including the Lat/Long/Elevation that defines the local origin location can be obtained through a call to `C3DRiClientPointData::GetPointInfo()`. If no navigation data is provided to the system (see Navigation Data below), then the points will be relative to the sensor origin.

### 7.5.6 Navigation Data Source

Observers can also be registered to receive navigation data from the 3DRi system. Use `C3DRiClientOPAL2::RegisterNavObserver()` with the name of the channel containing nav data to be observed and provide an observer object derived from the `I3DRiClientNavDataObserver` interface to handle incoming navigation data. See the Doxygen documentation for the contents of the `NavData_t` structure.

### 7.5.7 External Inputs

The 3DRi API includes the ability to provide external navigational information to improve system accuracy when geo-referencing the point data.

#### 7.5.7.1 Navigation Data

Navigation information from an external INS (inertial navigation system) or GPS can be input using `C3DRiClientControl::UpdateNav()` which takes an `ExternalNavInput_t` structure as input. See the Doxygen documentation for a complete listing and description of the `ExternalNavInput_t` structure elements.
7.5.7.2. GPS Time Synchronization

To ensure that the internal timestamps of the point data match the external navigation data provided, the time from a GPS can be used to synchronize the sensor clock to the pulse-per-second (PPS) signal from the GPS.

Use C3DRiClientControl::EnableTimeSynchronization() to enable the time synchronization on either the rising or falling edge of the PPS signal.

After time synchronization is enabled, C3DRiClientControl::SetTime() should be called at a regular 1Hz interval with the current time as standard NEMA 0183 GPZDA formatted string obtained from the gps/ins system.
8.0 Maintenance and Service

There are no service parts or components on the OPAL sensor. Return any item suspected of failure or needing service to Neptec Technologies via the RMA process. To begin this process contact Neptec Support at ntcsupport@neptec.com

Manufacturing address:

Neptec Technologies  
302 Legget Drive, Suite 202 
Ottawa, ON K2K 1Y5

The OPAL sensor may require regular operational maintenance.

It is recommended that the OPAL sensor be inspected during use and if necessary clean the optical window. Having a clean unobstructed aperture window will ensure the OPAL sensor continues to operate at peak performance.

8.1 Maintaining a Clean Aperture window

**WARNING:** Ensure the OPAL sensor is unpowered to prevent any emitted laser beam during inspection.

During operations, the OPAL sensor should be visually inspected daily for cleanliness and, if required, should be cleaned with soap using a clean, soft, damp cloth. **Do not use solvents, dry cloths, paper, oily rags or scrubbing pads.**

The OPAL sensor is designed to be pressure washed when needed. Ensure the pressure washer nozzle does not come closer than 1 foot from the sensor. Do not use aggressive detergents. The OPAL Conical window is sapphire glass and depending on the performance grade may have an additional anti-reflection coating applied. The OPAL Panoramic dome is Acrylic.
9.0 Troubleshooting

9.1 The optical window is damaged

If the OPAL Sensor’s optical window has been damaged, it must be replaced. Contact Technical Support for details on replacements.

9.2 The LEDs on the OPAL Sensor do not come on

When the LEDs are on it indicates that the laser has the ability to fire. (It is not an indication that the laser is necessarily firing, merely that it is ready to fire.) If, several minutes after power-up (and after the laser safety tests have completed), the LEDs are not on, check to see if there are any error messages indicating a sensor initialization failure (such as “Unable to set motor speed,” or “Unable to set laser power”). A check for these messages should also be performed if the LEDs turn off during normal operations. If any such messages are present, contact Technical Support.

9.3 Ethernet communications with sensor are not available

If you have been previously able to successfully establish communications between System Manager or the Viewer and the sensor, check that the Ethernet cable connecting the control computer to the OPAL sensor is secure at both ends. If the physical cabling connection seems secure, verify that the Ethernet link between the control computer and the sensor is valid using the ‘ping’ command. (At the command prompt on the control computer, type the command ‘ping <sensor IP address>’. If the link is valid, messages will be displayed stating that packets of data have been received back from the IP address specified.)

9.4 The motor in the OPAL Sensor is not spinning

If the motor is not spinning, check that power is applied correctly to the OPAL sensor (and that the power supply is functioning correctly by using a multi-meter to measure the voltage output by the power supply). If power is applied correctly to the sensor, the next step is to look for any failure messages from the sensor. If a failure message has been reported, contact Technical Support for directions on how to proceed.

9.5 OPAL Sensor time synchronization not working

If the API reports a problem stating that time synchronization has failed, make sure that the Navigation System you are using is outputting a Pulse Per Second (PPS) signal. Next, make sure that there is a cable delivering the PPS to the OPAL sensor. Finally, verify that the Navigation System is received data from the GPS.

9.6 No scan data points are returned

If a scan appears to complete successfully but no scan data points are returned: Check to verify that the Laser Power in the scan command is not set to zero. (The Laser Power is one of the Scan Parameters.) If the above is OK, check for failure messages being reported by the OPAL.
sensor. If any such messages have been reported, contact Technical Support for further directions NTCSupport@neptec.com.
10.0 Service and Warranty information

Neptec warrants each Product(s) unit purchased by you against defects in workmanship and materials for a period of one (1) year after the date of delivery to you. For details, please refer to our Product Purchase Terms and Conditions available on our website at

http://www.neptectechnologies.com/downloads/

To return a product to service, please contact your sales representative or Neptec customer support for instructions to obtain a Return Merchandise Authorization (RMA) number.

http://www.neptectechnologies.com/contact-us/
11.0 Appendix – OPAL Installation Control Diagram (ICD)
NOTES:
1. APPLICABLE STANDARDS / SPECIFICATIONS:
   A. ASME Y14.10
   B. ASME Y14.5M-1994
2. REFS:
   MATERIALS AND PROCESSES USED IN THE MANUFACTURING OF THIS PART MUST CONFORM TO EU DIRECTIVE 2000/53/EC AND ALL ITS AMENDING DIRECTIVES AT DATE OF PURCHASE ORDER. NO SUBSTITUTIONS PERMITTED WITHOUT PRIOR WRITTEN APPROVAL.
4. PERMANENTLY MARK "POD001-S1-XX-BY-2", WHERE "XX" DENOTES HARNESS LENGTH PER TABLE 2 ON CABLE HARNESS IN APPROPRIATE AREA USING A SELF-LAMINATING THERMAL TRANSFER POLYVINYL FLUORIDE OR ENGINEERING APPROVED EQUIVALENT LABEL.
5. LABEL EACH CONNECTOR USING A SELF-LAMINATING THERMAL TRANSFER POLYVINYL FLUORIDE OR ENGINEERING APPROVED EQUIVALENT LABEL.
6. CONNECTOR ASSEMBLY TO BE WATER-PROOF SEATED, USE POTTING COMPOUND TO ENSURE WATER-PROOF SEAL AROUND CONNECTOR BACK-HALF (ITEM 3), AND CABLE (ITEM 3).