

# OpenGC Programmer's Guide

Damion Shelton

## 1 Licensing

### 1.1 Software

OpenGC is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but *without any warranty*; without even the implied warranty of *merchantability* or *fitness for a particular purpose*. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.

OpenGC is intended for entertainment purposes only. The user accepts all liability that may arise from installation or use of this software.

### 1.2 Developer's guide

This document is *not* distributed under the GPL. It is ©2003 by Damion Shelton. You are free to redistribute the unmodified PDF form of this guide in electronic form, but may not make modifications without permission. Like OpenGC, this guide is a work in progress...so please understand that there may be omissions. For extra assistance, send email to [opengc-devel@lists.sourceforge.net](mailto:opengc-devel@lists.sourceforge.net)

## 2 Style guide - read me before coding!

Please adhere to the following style conventions when writing code for OpenGC. These conventions help to ensure easy interpretation of code in OpenGC and provide for consistent formatting between developers.

### 2.1 Class and variable names

- Variable and class names should be as verbose as necessary to describe the intent of the variable/object without needing to wade through a morass of acronyms. The exception to this is acronyms which are accepted as common aviation language, for instance PFD, EICAS, etc. For example, `m_PFDskyColor` is an acceptable variable name, `m_Hyk8` is not.
- In general, all class and variable names should follow the “camel hump” style of capitalization - *thisIsCamelHumpCapitalization*

- Variables which are class members should be named using the Microsoft convention of prefacing the variable name with `m_`. Variables which are local to a function do not have the `m_` preface and do not have their first character capitalized. As an example, indicated airspeed in knots would be named `m_IndicatedAirspeedKnots` if a class member and `indicatedAirspeedKnots` if a local function variable. Certain functions and classes may break this convention; for instance, the FSUIPC data source names raw values with a `_` character preface.
- Classes which are part of OpenGC should be prefaced with *ogc*, for instance *ogcMyClass*.

## 2.2 Tabs and indentation

- Configure your editor to insert 2 spaces rather than hard tabs. This ensures that indentation appears constant regardless of the tab setting of the editor being used.
- Indent function/class/loop/etc. braces flush with the line immediate above, and then indent the body of the function one additional level. For instance:

```
for(int i=0; i<10; i++)
{
    cout << i << endl;
    // a bunch of
    // code here
}
```

## 2.3 Comments

Please comment liberally, and explain anything that is not immediately obvious. When commenting class header files, please use doxygen style comments, e.g. `/** This is a comment */` to permit automatic generation of documentation web pages. When commenting code elsewhere, please use `//` rather than `/* */` - this makes it easier to comment out large sections of code when debugging.

## 2.4 Headers and copyrights

All classes which are part of OpenGC should include the following copyright/licensing header at the top of the file. Material between the “\$” marks is filled in by CVS to indicated revision history. You should modify the original author and contributors as appropriate.

```
/*=====

OpenGC - The Open Source Glass Cockpit Project
Please see our web site at http://www.opengc.org

Module:  $RCSfile: ogcNavDatabase.cpp,v $

Copyright (C) 2001-2 by:
    Original author:
```

Damion Shelton  
Contributors (in alphabetical order):

Last modification:

Date:           \$Date: 2003/06/04 14:33:18 \$  
Version:       \$Revision: 1.5 \$  
Author:        \$Author: damion \$

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

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

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.

=====\*/

Classes and/or other files which are distributed with OpenGC but were not created by the OpenGC team (e.g. SimGear) should preserve whatever formatting and licensing was distributed with the original source code. If appropriate, include in CVS a copy of any licenses or notices that accompany the code.

### 3 OpenGC modules

OpenGC is designed in a modular, object-oriented manner in order to facilitate reuse of code. There are several main components to OpenGC, and it's important to understand what each does when designing a new gauge.

1. **main.cpp** - Although this file contains the `main()` function, very little actually occurs here. Several global objects are created, notably the primary executable object `ogcAppObject`. After instantiating these objects, a rendering callback is established in FLTK and OpenGC enters a rendering loop.