VisBox User Guide

This guide will take you through the basics of configuring, running, and developing on the VisBox system.

Turning It On

1) Turn on projectors

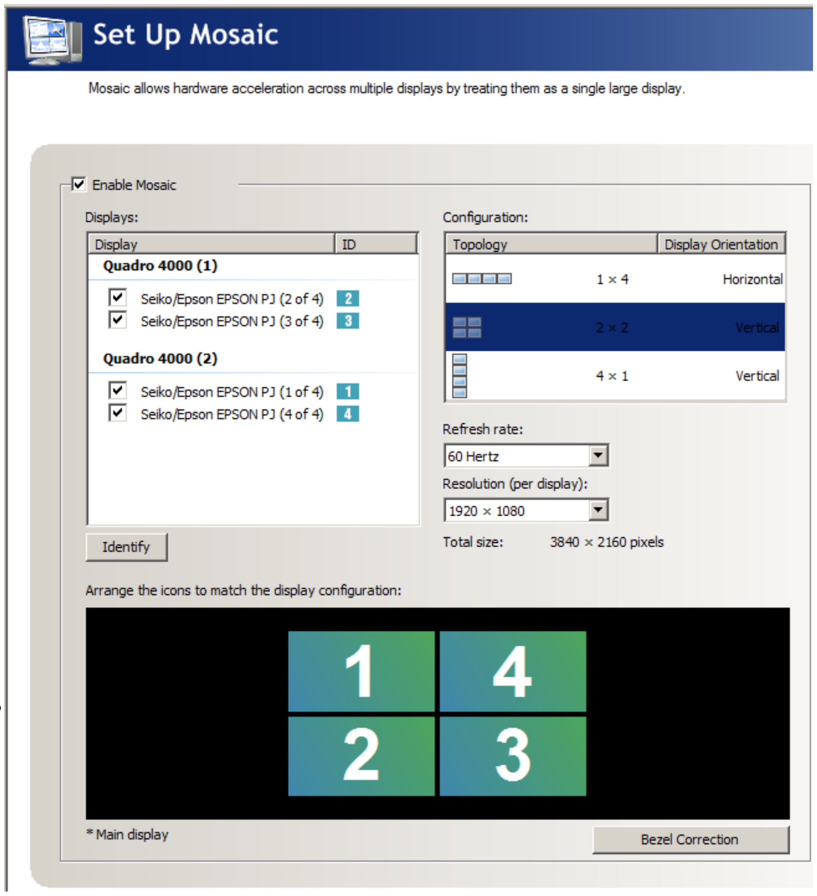
An EPSON remote control is provided to turn the projectors on and off. The four projectors receive the signal individually, so its important to hold down the power button for several seconds so they all see it.

When a projector is on, it will have two blue lights. When it's off, there will only be one blue light. If you end up with a situation where some projectors are on and some are off, you can manually push a projector's power button.

Turning the projectors off is similar, but you should push and hold the remote control's power button twice.

2) Turn on computer

The VisBox computer is a black extended-length tower. There's a power button on the front.


Set Up Mosaic

Mosaic allows hardware acceleration across multiple displays by treating them as a single large display.

☑ Enable Mosaic

Displays:

Display	ID
Quadro 4000 (1)	
☑ Seiko/Epson EPSON PJ (2 of 4)	2
☑ Seiko/Epson EPSON PJ (3 of 4)	3
Quadro 4000 (2)	
☑ Seiko/Epson EPSON PJ (1 of 4)	1
☑ Seiko/Epson EPSON PJ (4 of 4)	4

Configuration:

Topology	Display Orientation
1 × 4	Horizontal
2 × 2	Vertical
4 × 1	Vertical

Refresh rate: 60 Hertz

Resolution (per display): 1920 × 1080

Total size: 3840 × 2160 pixels

Arrange the icons to match the display configuration:

The Desktop

There are four projectors; two behind the screen, and two above the VisBox floor. Each pair of projectors cover the same surface area. Each of the back projectors covers the entire screen, and each top projector covers the entire floor. They have polarized filters that create the proper image for the user's left and right eyes.

Windows interprets the cameras as being part of one enormous desktop, and doesn't understand that the cameras overlap. If you move the cursor to past the right edge of the screen, it will reappear on the left edge. If you can't click on something, you may be in the wrong (but overlapping) half of the desktop.

To access the desktop configuration, right-click on the desktop and select "NVidia Control Panel". Then go to Workstation->Set Up Mosaic. NOTE: Modifying the configuration will prevent stereo 3D applications from running properly.

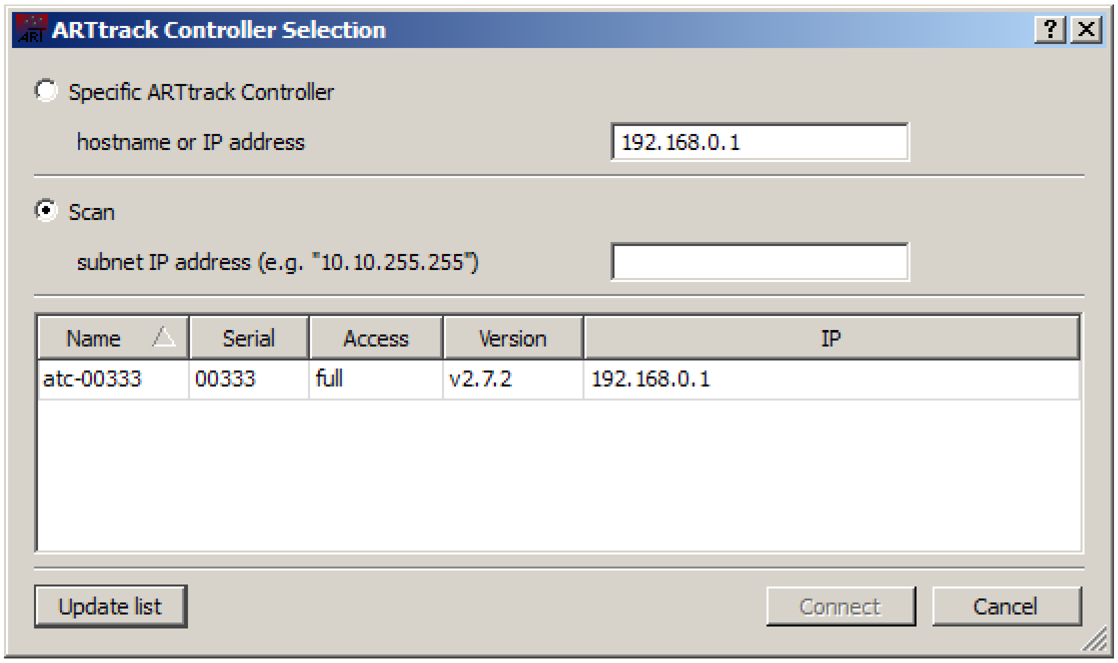
Start DTrack2

There are two cameras mounted on top of the screen that were included with the VisBox system. The cameras can be accessed through Desktop->DTrack2.

When you first start DTrack2, a small window appears in the 'center' of the huge virtual desktop. You can move it to a visible position with the 2, 4, 6, and 8 keys. Hit the 'Connect' button, or press Enter, to connect to the ART Tracker.

If hitting Enter instead brings up an 'ArtTrack Controller Selection' window, there was a problem connecting to the ART box. Things to try:

1) There are two radio buttons, 'Specific ArtTrack Controller' and 'Scan'. Try hitting 'Specific ArtTrack Controller' followed by 'Scan' to automatically detect the ART box. Click 'Connect' if it worked.



hostname or IP address: 192.168.0.1

[✓ Scan]

subnet IP address (e.g. "10.10.255.255"):

Name	Serial	Access	Version	IP
atc-00333	00333	full	v2.7.2	192.168.0.1

2) If that doesn't work, you can manually reboot the ART box. It's a black box labelled 'ART Track Controller /TP'. Just hold down the power button to shut it off, wait a few seconds, then press it again to turn it back on.

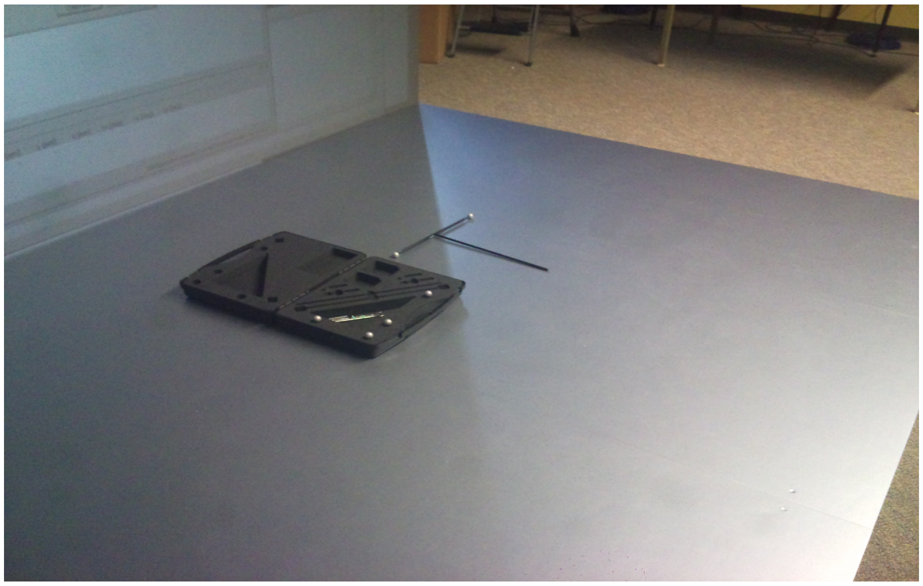
Calibrate Tracking

The cameras may lose their calibration if jostled or moved. This can prevent the wand and glasses' position data from being received by your applications, or reduce the accuracy of the tracking. To re-calibrate:

1) Find the ART calibration kit. It's a small black box with the ART logo on it.

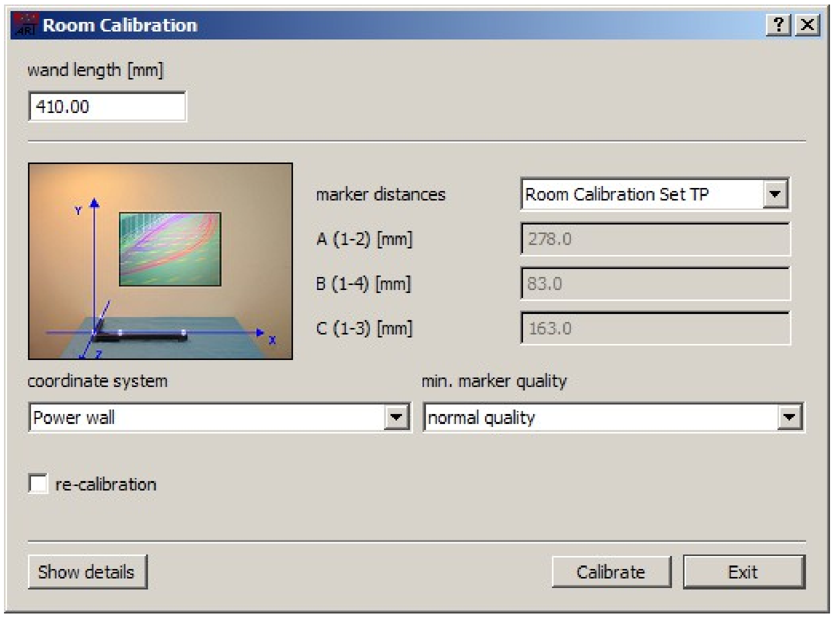
2) Assemble the wand. There's a rod with two reflective spheres at either end, and a place to attack a second rod in the center.

3) Open the calibration kit and place it in the center of the VisBox floor. It has reflective markers built in to it, which are used to orient the cameras. The markers make an 'L' shape - the long part of the L should point to the right, and the short part should point towards the screen. It's important for the box to be positioned very accurately, since this will affect the quality of the calibration.



4) Make sure no other reflectors are being picked up by the cameras - just the wand and calibration kit. You can check this in the main window of DTrack2.

5) Open DTrack2 and go to Calibration->Room. If it's greyed out, you may need to click on 'Stop' to stop tracking.


Room Calibration

wand length [mm]: 410.00

marker distances: Room Calibration Set TP

A (1-2) [mm]: 278.0

B (1-4) [mm]: 83.0

C (1-3) [mm]: 163.0

coordinate system: Power wall

min. marker quality: normal quality

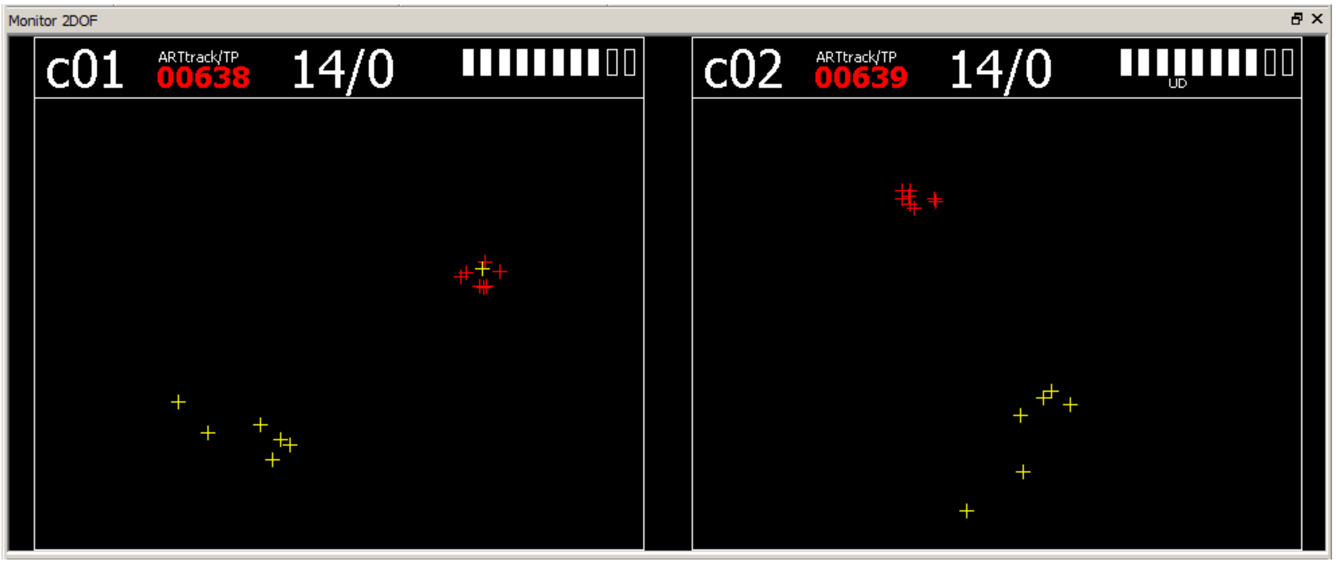
re-calibration

6) Hit 'Calibrate' to start calibration. While the kit is laying open on the floor, twirl the wand through the space tracked by the cameras.

You're done!

Start Tracking

In DTrack2, click on 'Start' to begin tracking. When the VisBox cameras are idle, they'll have one red light. When they're tracking, they'll have two. You should be able to see tracked reflectors in DTrack2's main window. If not, you can check the 'Event Display' section for any error messages.



If you see an error like "no connection to camera proxy" or "zero cameras found", you can try rebooting the ART box. This can be done through DTrack2 by going to DTrack2->ARTTrackController Reboot. If that doesn't work, you can try physically rebooting it.

When finished, always make sure to turn the cameras off - their infrared emitters will burn out if left on continually. (Make sure only one red light is present on the cameras.)

VRJuggler

We use a framework called VRJuggler to develop and run 3D applications. It's a complex framework that incorporates many other libraries. These instructions assume that everything is installed in C:\tools, and you will need to modify the file paths if VRJuggler or its dependencies are installed in a different location.

Configuration Files

VRJuggler has multiple configuration files, located at C:\tools\config\juggler3.0. They should also be included with this documentation. The config files define the position, orientation and size of the displays, and any input devices that are being used. VRJuggler applications will want to access c2-hd.jconf file, which in turn references the others. Therefore, all of the .jconf files should be in the same directory as your executable.

Environment Variables

OSGHOME=C:\tools\OpenSceneGraph\OpenSceneGraph-3.0.1

OSG_FILE_PATH=C:\tools\OpenSceneGraph\OpenSceneGraph-Data-3.0.0

OSGPATH=%OSGHOME%\bin\osgPlugins-3.0.1;%OSGHOME%\bin;C:\tools\OpenSceneGraph\dep-x64\bin

VJ_BASE_DIR=C:\tools\VRJuggler\VRJuggler-3.0.1-1

VJ_DEPS_DIR=C:\tools\VRJuggler\VRJuggler-3.0.1-1-deps

VJ_CFG_PATH=%VJ_BASE_DIR%\share\vrjuggler\data\configFiles

VJPATH=%VJ_BASE_DIR%\lib;%VJ_DEPS_DIR%\bin;%VJ_DEPS_DIR%\lib;%VJ_BASE_DIR%\lib\gadgeteer\drivers;%VJ_BASE_DIR%\lib\gadgeteer\plugins;%VJ_BASE_DIR%\lib\jccl\plugins;%VJ_BASE_DIR%\lib\vrjuggler\plugins

Add %VJ_BASE_DIR%\lib;%VJ_DEPS_DIR%\lib to Path

Notes

These binaries are already on our system! No need to re-download them or set any environment variables.

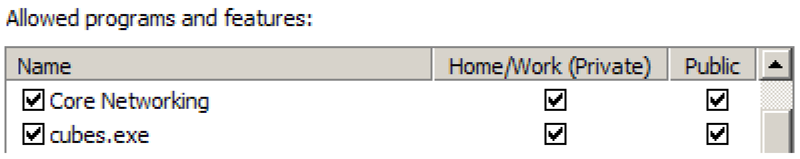
The exact values of the environment variables will differ depending on where the binaries are located.

Reboot after setting environment variables. This seems to be required since the variables reference each other.

Troubleshooting VRJuggler

I get a stream of errors in the console like: "Stable buffer is empty. If this is not the first read, then this is a problem."

This happens when the program can't get information from DTrack2. Ensure that DTrack2 is running and started. Otherwise, it's probably a firewall issue. When a VRJuggler program runs for the first time, Windows asks you if you'd like to unblock it. This doesn't seem to work, so you should access the firewall through the Control Panel and manually allow the program to access the network.


Allowed programs and features:

Name	Home/Work (Private)	Public
✓ Core Networking	✓	✓
✓ cubes.exe	✓	✓

Make sure both the private and public networks are checked, as well as the box next to the program name. You will need to do this every time the executable name changes, and when the executable is moved to a different directory.

I get a console error about the Pending list being stale.

If you see output like this:

```
[0003608/000] JCCL-RCFG:Pending list is now STALE:
                        0 items are still in the pending list
[0003608/000] DBG:---- Pending list: 0 items ----
                   ----------------------------------
```

There's probably an error in your configuration (.jconf) files. Be sure that you're using c2-hd.jconf, and that the other configuration files are present in the same directory.

No position data for the wand or glasses is being received, but I'm not getting any console errors.

The first step is to make sure that DTrack2 can see the objects you want to track. Next, check that your program can receive wand button input - that will let you know if your program is connecting to DTrack2. If you're getting button presses but no position data, re-calibrating DTrack2 may solve the problem.

I get an error when I try to run my program: Microsoft C++ exception: cppdom::Error at

memory location 0x0018f2cc..

This happens when cppdom is having problems reading the VRJuggler configuration files. Make sure that all of the .jconf files from c:\tools\config\juggler3.0 are in the same directory as your executable.

Developing with VRJuggler

Creating a New Project

VRJuggler requires some arcane project options that are almost impossible to successfully reproduce! It's far easier to take an existing solution and rename it than to create a new solution from scratch. There should be a sample project called HoloLoader along with this documentation – you can rename it to whatever you'd like.

Library Dependencies

HoloLoader interacts with four main libraries:

VRJuggler: Provides perspective correction based on the location of the user's glasses, and stereo 3D

*MathHelper***:** Math functions to rotate points and manipulate 3D objects.

EbonGL: Has a 3D scene that contains each object to be drawn. Includes drawing primitives such as cubes and spheres, controls lighting and shadows,

HoloLib: Higher level library, contains classes for loading 3D models from files, navigating through a 3D environment, and obtaining wand input.

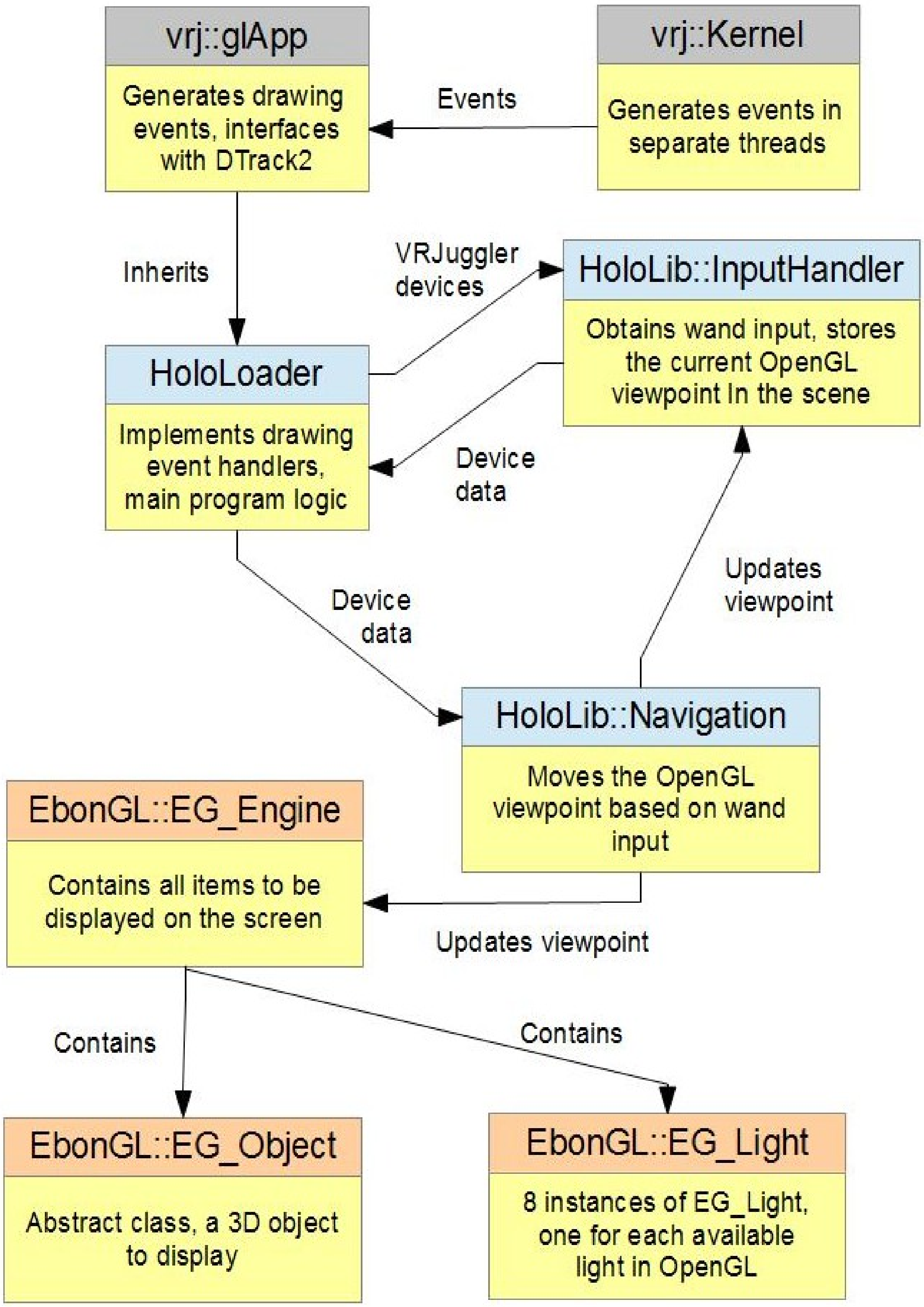
These libraries in turn rely on other libraries, making a hierarchy of dependencies. All libraries should be located in c:\tools. All libraries are 64-bit to allow us to take full advantage of the RAM on the VisBox computer.

main.cpp

HoloLoader's main.cpp file is the starting point into the program. It creates a VRJuggler Kernel that generates events for the rest of the program, and loads the configuration files that tell VRJuggler about the wand, reflective glasses and screen layout. (See Configuration Files in the VRJuggler section.)

HoloLoader.hpp

HoloLoader is a class that inherits from VRJuggler's GlApp class. It contains the logic for the program, and implements drawing event handlers inherited from GlApp.


vrj::glApp
Generates drawing events, interfaces with DTrack2
vrj::Kernel
Generates events in separate threads
Events
Inherits
VRJuggler devices
HoloLib::InputHandler
Obtains wand input, stores the current OpenGL viewpoint In the scene
HoloLoader
Implements drawing event handlers, main program logic
Device data
Updates viewpoint
Device data
HoloLib::Navigation
Moves the OpenGL viewpoint based on wand input
EbonGL::EG_Engine
Contains all items to be displayed on the screen
Updates viewpoint
Contains
Contains
EbonGL::EG_Object
Abstract class, a 3D object to display
EbonGL::EG_Light
8 instances of EG_Light, one for each available light in OpenGL