VtdApi

The **VtdApi** aims to offer a C++ interface to control and interact with VTD and it's simulation via RDB and SCP in a convenient way.

About

VtdApi offers convenience functions for simulation control both via RDB and SCP. It relies on VtdFramework, which is a common codebase for VTD and accompanying samples and tools. Ultimately, it's aim is to increase type safety and decrease the effort required to write code for VTD while providing a common interface throughout.

Features

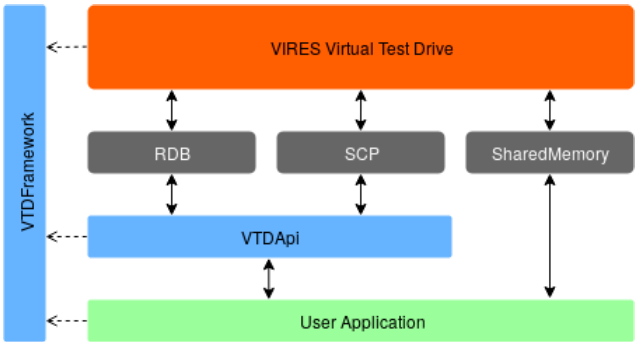
VtdApi offers:

- Send/Receive SCP Messages to a single host (per VtdApi instance)
- Send/Receive RDB Messages to/from multiple RDB connections and hosts
- Simulation control (Configure, Apply, Start, Run, Pause, Stop, Change Project & Setup, etc.)
- Keep track of VTD Operating Stage and process running states.
- Type and syntax safe building of SCP Commands

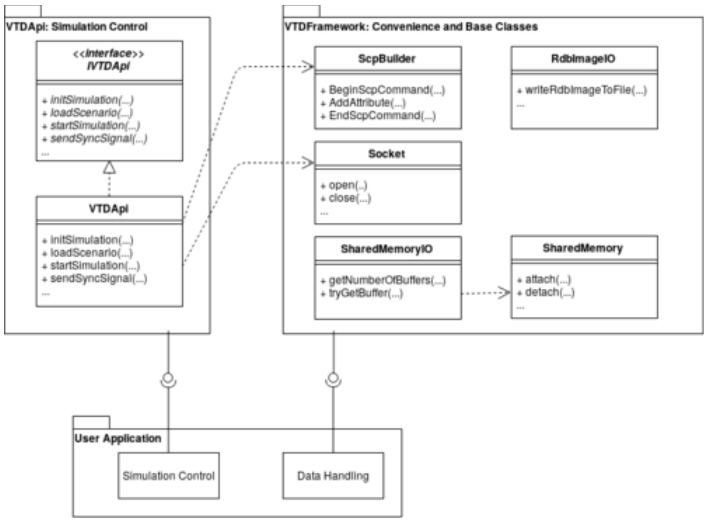
Additionally the VtdFramework libraries offer:

- Convenient Shared Memory access
- Storing RDB Images to files
- C++ Event Handler

SW Landscape



Structure



Sample Project using CMake

VtdApi comes with CMake configuration files allowing one to set up a build quickly while hiding the details. The following is a simple `CMakeLists.txt` which makes use of CMake as a makefile generator. Paths for include files and the library are set up automatically, only the path to the cmake configuration files needs to be set.

```
cmake_minimum_required(VERSION 3.5)

set(CMAKE_PREFIX_PATH "./lib/VtdApi/cmake" CACHE STRING "") #this line tells cmake where it can find the configuration files
find_package(VtdApi CONFIG REQUIRED) #this line tells cmake to set up VtdApi

project(MyProject VERSION 1.0.0)

add_executable(MyProgram main.cpp)

target_link_libraries(MyProgram VtdApi VtdFramework)
```

VtdApiSamples

VtdApiSamples offers several examples of VtdApi usage. They show how RDB and SCP messages can be received, after receiving created appropriate message, and send back to the Virtual Test Drive. The process of automating VTD has been also illustrated in several examples. In addition examples include the creation of a simple adaptive drive control and its use.

Example Basic

Basic simulation control. (VTD started manually, Standard setup and SampleProject).

ExampleScpBuilder

Basic simulation control with custom scp message to set speed. (VTD started manually, Standard setup and SampleProject)

ExampleShm2Png

Simulation control with sync triggering via RDB and reading SHM buffer.

Enable RDB sync trigger and set SHM keys in your setup.

The numbers of SHM keys should match the numbers in the setups. Last parameter should match buffer release flag from setup.

Usage:

Shm2PngDual 2 0x0120a 0x0120b 2

ExampleTestAutomation

Full automation, uses Sample Project and Standard and executes scenarios.

vtdStart.sh requires you to set the VTD_ROOT environment variable.

Assuming VTD is at ~/VTD and you are using bash or zsh:

```
> export VTD_ROOT=~/VTD
```

Usage:

```
> ./ExampleTestAutomation $VTD_ROOT
```

Alternatively you can specify an alternative setup:

```
> ./ExampleTestAutomation $VTD_ROOT &lt;setup_name&gt;
```

ExampleConsole

ExampleConsole defines - the entry point for the console application.

It receives RDB message, process it, print data to console and send trigger via RDB message.

Set simulation control with sync triggering via RDB.

ExampleConsoleDriverCtrl

ExampleConsole defines - the entry point for the console application.

It receives RDB message, procces it, print data to console and send simple driver control message.

Ego's car increases speed and decreases based on sine function and simulation time.

ExampleConsoleDriverCtrTorque

ExampleConsole defines - the entry point for the console application.

It receives RDB message, procces it, print data to console and send simple driver control message.

Ego's auto increases and decreases speed, and applies torque based on sine function and simulation time.

ExampleConsoleProxyPkg

ExampleConsoleProxyPkg defines - the entry point for the console application.

It receives RDB message, procces it, print data Proxy to console and send proxy message to VTD.

ExampleDriveControl

Sample showing how to use VtdApi with RDB Messages, and simple ACC which breaks when an object has been detect.

Start VTD with Standard setup and Sample project.

Set sensor properties.

Sample receives RDB message, process it, print driver data to console and breaks when objects is inside sensor range.

ExampleDriveControlAdvanced

ExampleDriveControlAdvanced is an example how to use VtdApi to control vehicle.

1. It receives data from sensor,
2. Calculate position of detected object,
3. Check if the object has same direction as Ego,
4. Control ego's vehicle to follow object in front of him and to keep same speed and distance.

Start VTD with Standard setup and Sample project.

Set sensor properties.

ExampleVehDynInteq

ExampleVehDynInteq is an example how to use VtdApi to fully control vehicle.

It process object state, calculate dummy object state (only moving object in x direction) and send object state to VTD.

Start VTD with Standard setup and Sample project.

ExampleVehDynInteqUDP

ExampleVehDynInteqUDP is an example how to use VtdApi to fully control vehicle.

It process object state, calculate dummy object state (only moving object in x direction) and send object state to VTD.

Start VTD with Standard.UDP setup and Sample project.

This only example showing receiving and proccesing UDP message.

FAQ

Why are VtdApi and VtdFramework separate?

VtdApi offers high-level functionality that is useful for application code only, whereas VtdFramework offers a common codebase for different VTD modules - one of which is VtdApi.

VtdApi_Structure.png (11.4 KB) Bernhard Bieder, 21.12.2018 09:52
VtdApi_UML.png (50.8 KB) Bernhard Bieder, 21.12.2018 09:52