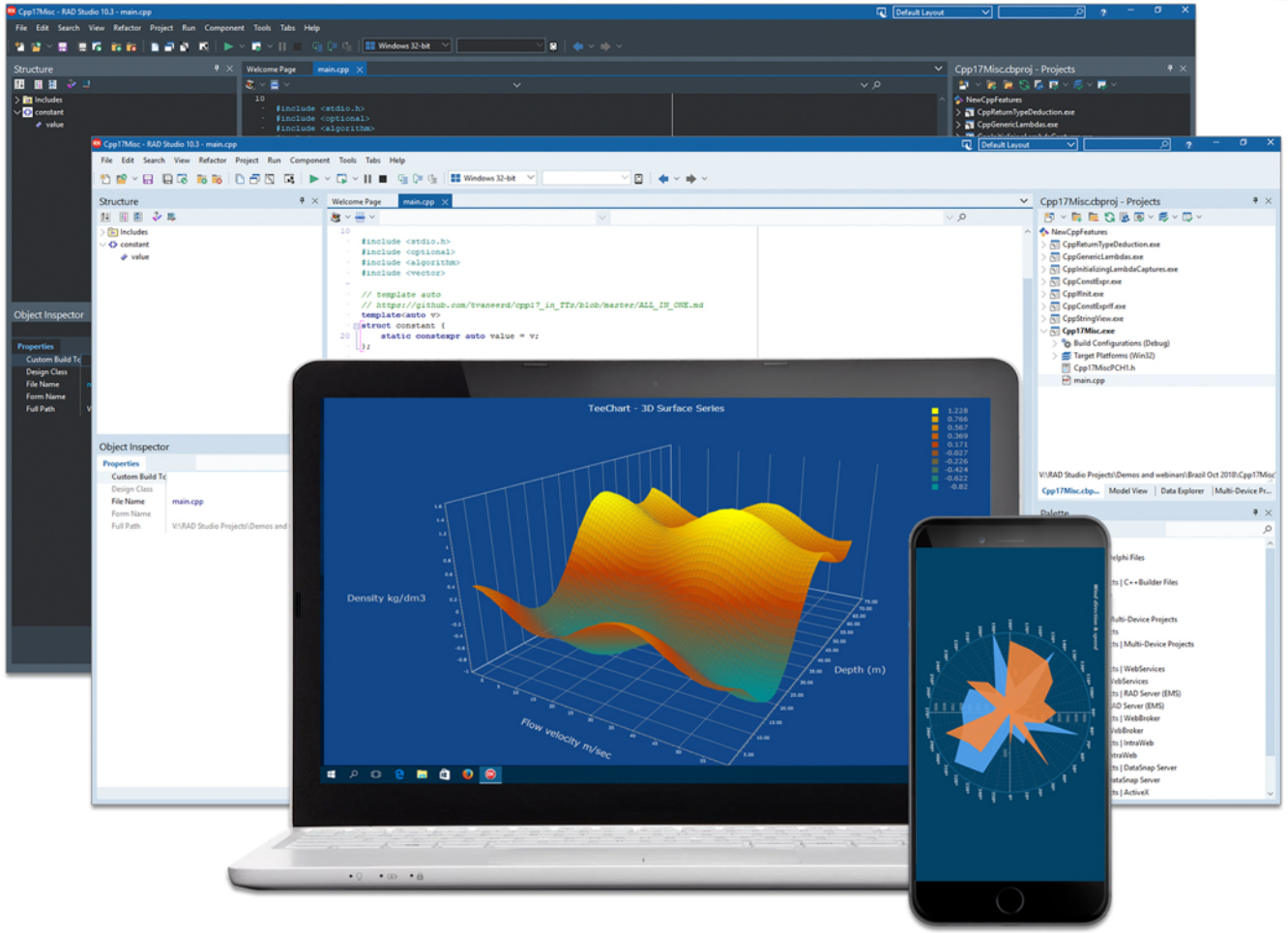
embarcadero®

C++ Builder



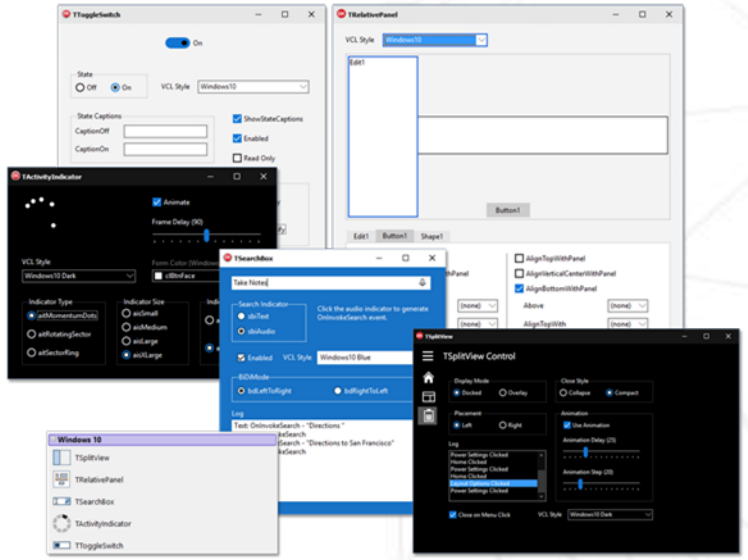
Al Mannarino
Embarcadero Engineer

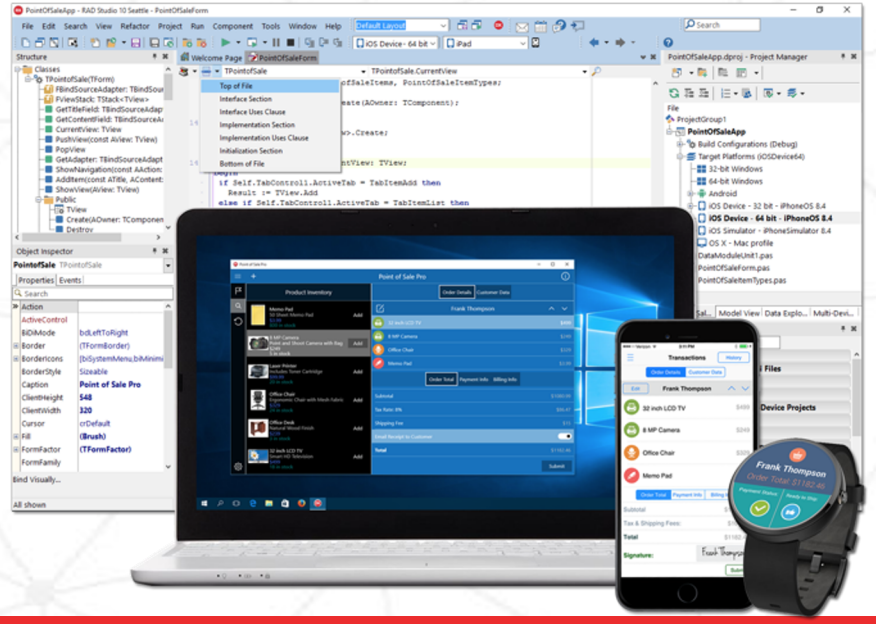
embarcadero®

Agenda

1. C++ Builder Overview

2. C++ Builder Levels of Development:
    1. Components (Visual Component Library (VCL) / FireMonkey (FMX))
    2. **Common Libraries** (Run Time Library (RTL))
    3. **Platform APIs** ('Touch the Metal')

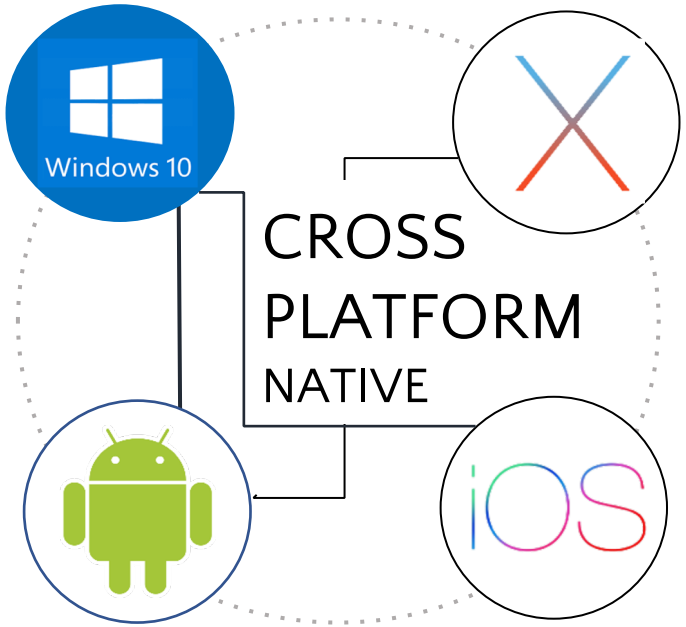


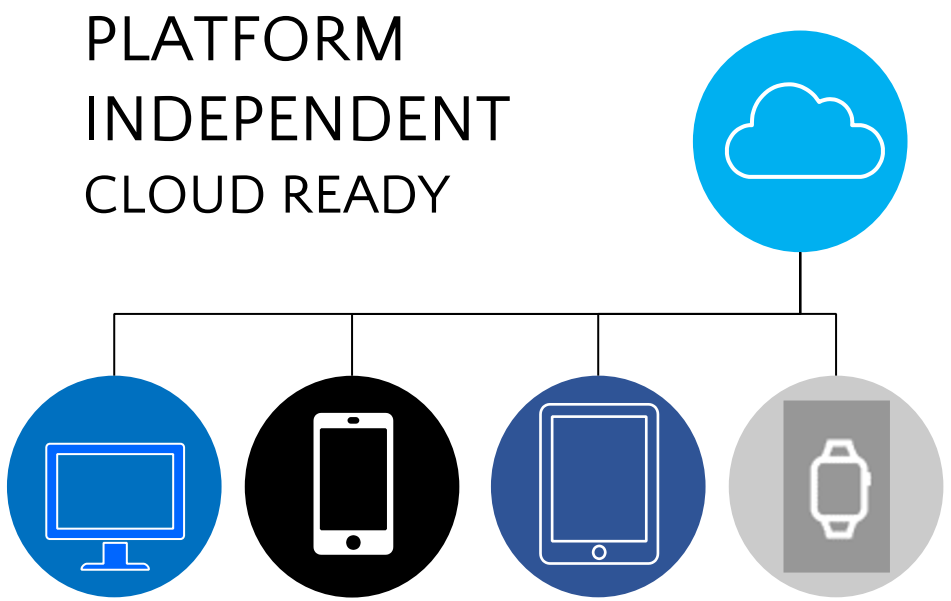


embarcadero®

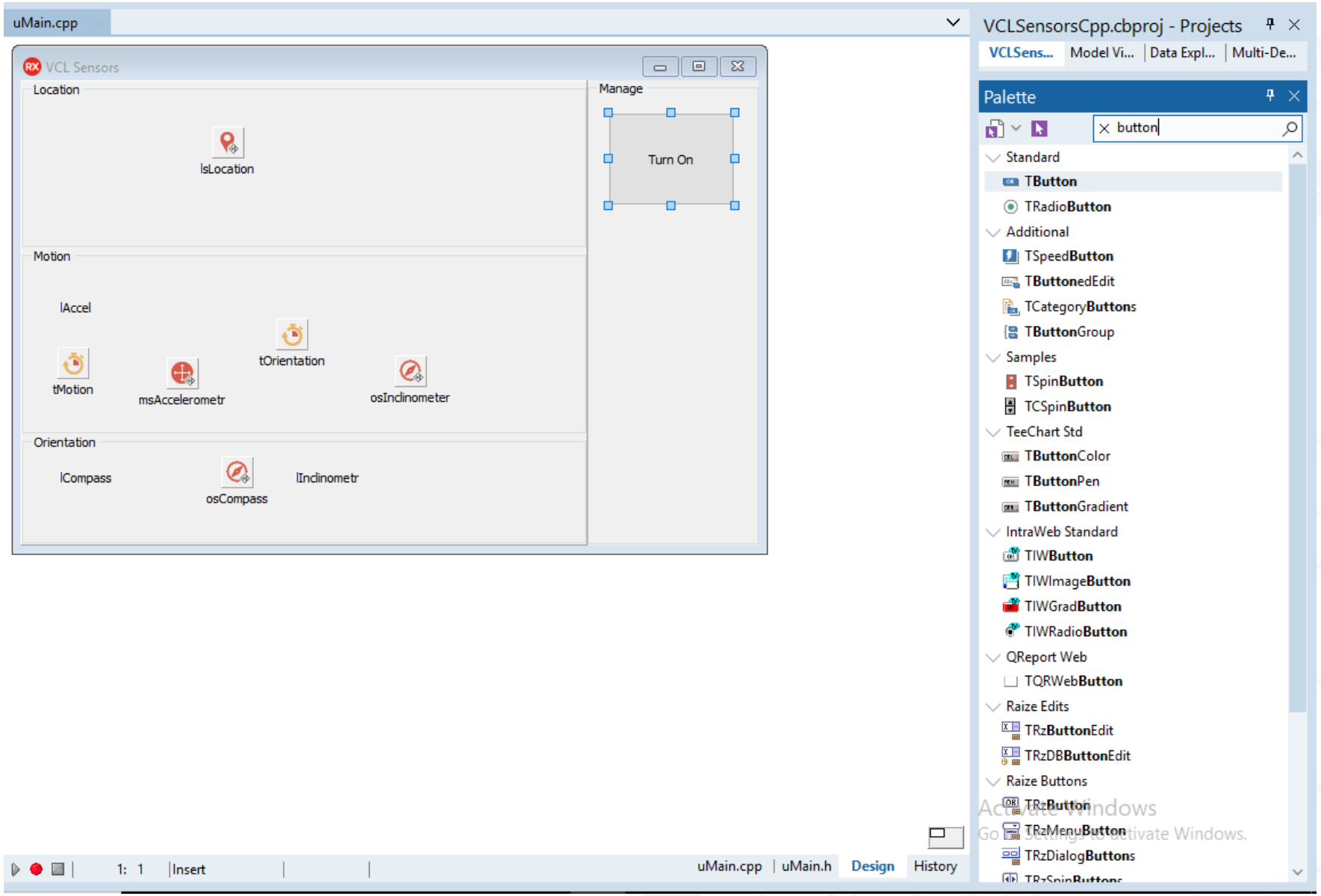
What is C++ Builder?

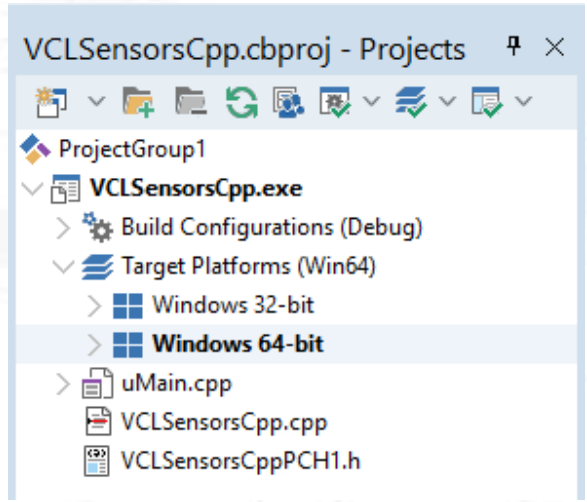
C++ Builder is both a **C++ IDE plus a Full Stack** for application and server development, with features C++ developers love: **design, code, debug, and test for cross-platform deployment with native performance.**



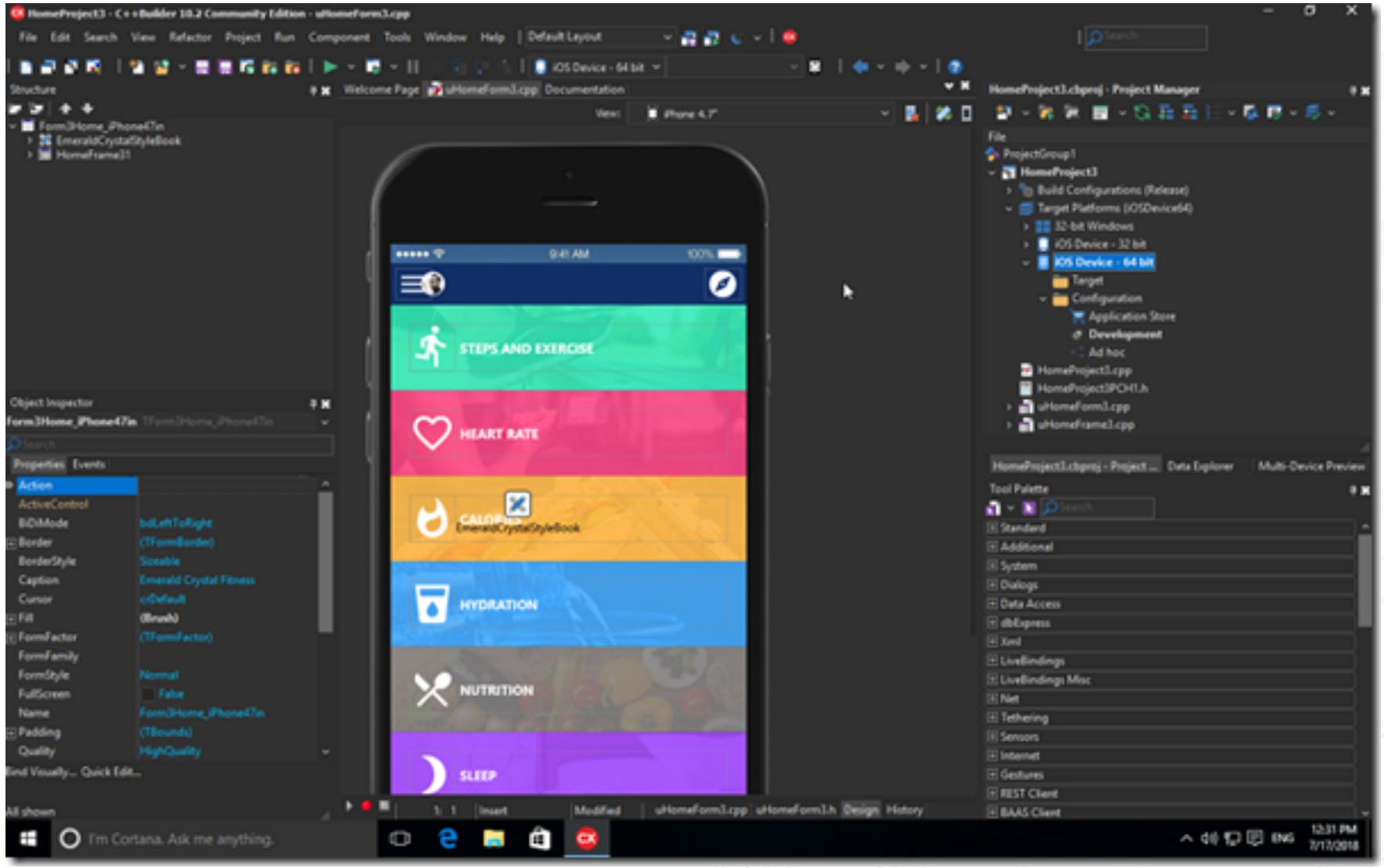


Components – Visual Component Library (VCL)





Components – Multi-Device (Firemonkey or FMX)

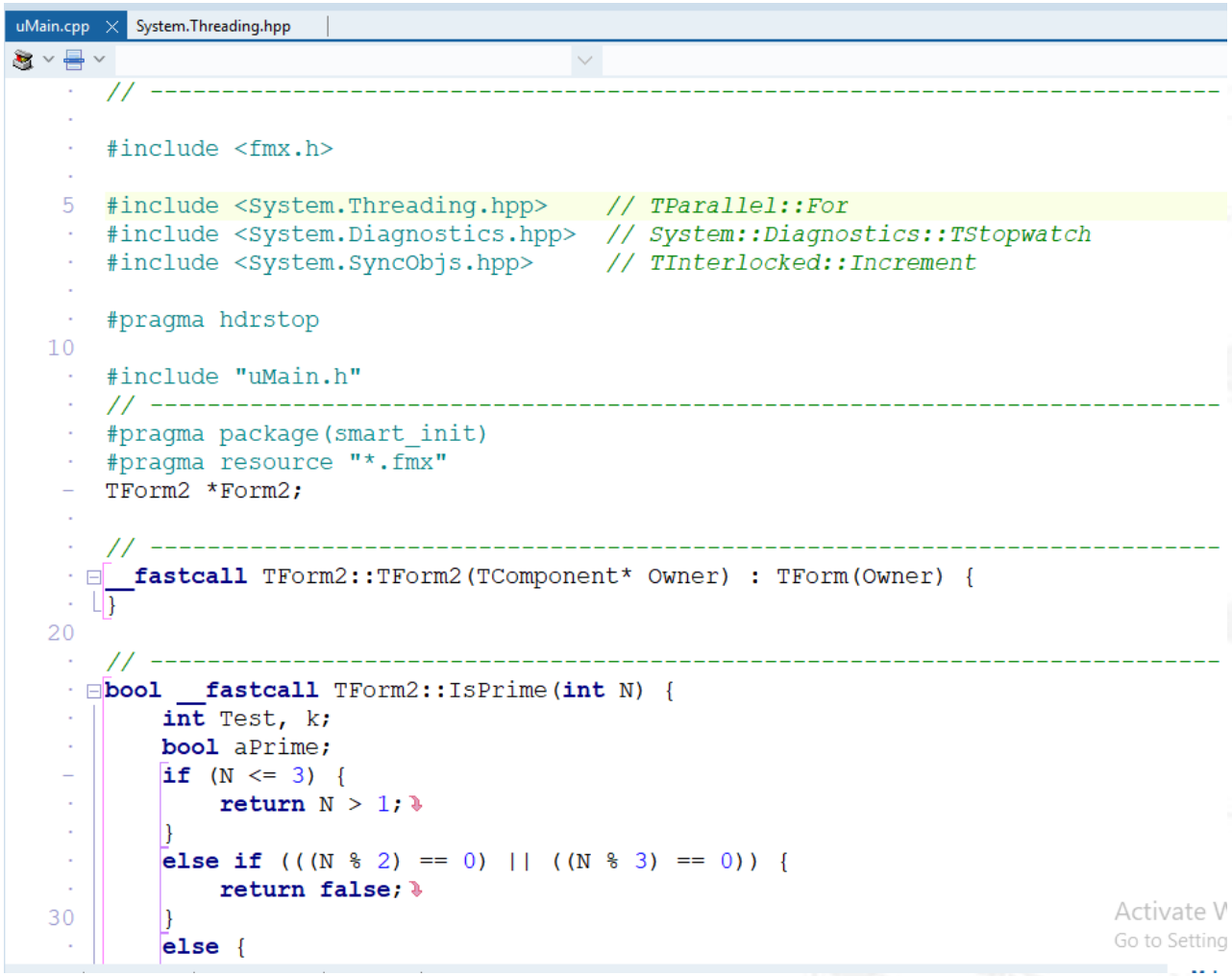


PointOfSaleApp.cbproj - Projects

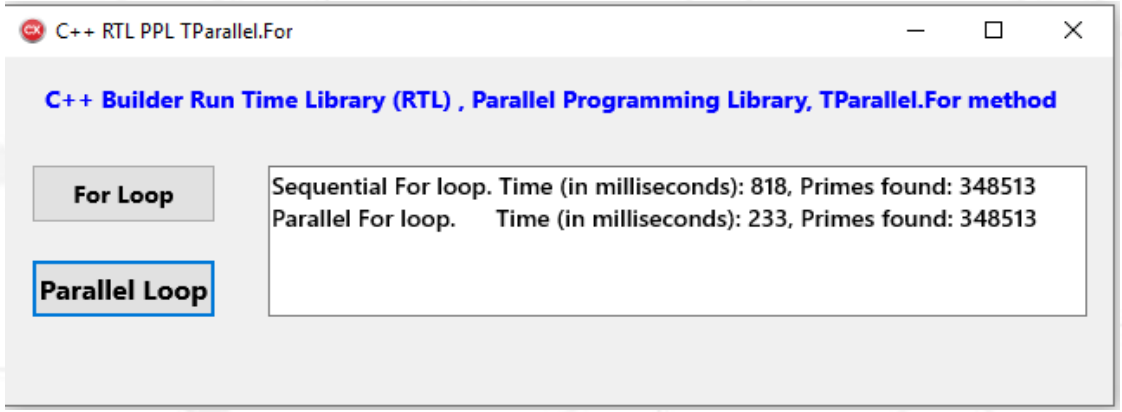
- ProjectGroup1
- **PointOfSaleApp**
  - Build Configurations (Debug)
  - Target Platforms (iOSDevice64)
    - Android 32-bit - Android SDK 25.2.5 32-bit
      - Target
        - Nexus 7
      - Configuration
      - Libraries
    - iOS Device 32-bit - iPhoneOS 13.1 - MacOSX profile
      - Target
        - admin's iPad
      - Configuration
    - **iOS Device 64-bit - iPhoneOS 13.1 - MacOSX profile**
      - Target
        - **admin's iPad**
      - Configuration
    - macOS 32-bit - MacOSX 10.15 - MacOSX profile
    - Windows 32-bit
    - Windows 64-bit
  - MasterDetailTablet_Search.cpp
  - PointOfSaleApp.cpp
  - TabletMasterDetailwithSearchPCH.h

embarcadero®

Common Libraries (RTL)




```
#include <System.Math.hpp>
#include <System.Bluetooth.hpp>
#include <System.Sensors.hpp>
#include <System.Threading.hpp>
```



Platform APIs

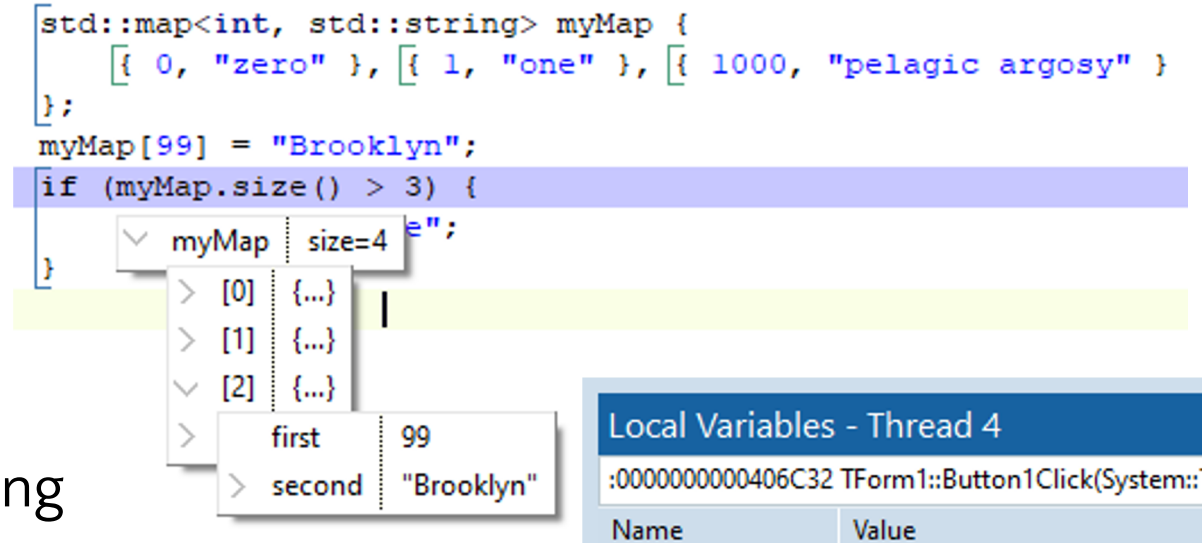
```
#include <fmx.h>
#if defined(TARGET_OS_IPHONE) || defined(TARGET_IPHONE_SIMULATOR)
#include <iOSapi.UIKit.hpp>
#endif
#pragma hdrstop

#include "uMain.h"
#if defined(__ANDROID__)
#include <Androidapi.JNI.Os.hpp>
#include <Androidapi.Helpers.hpp>
#endif
//
```

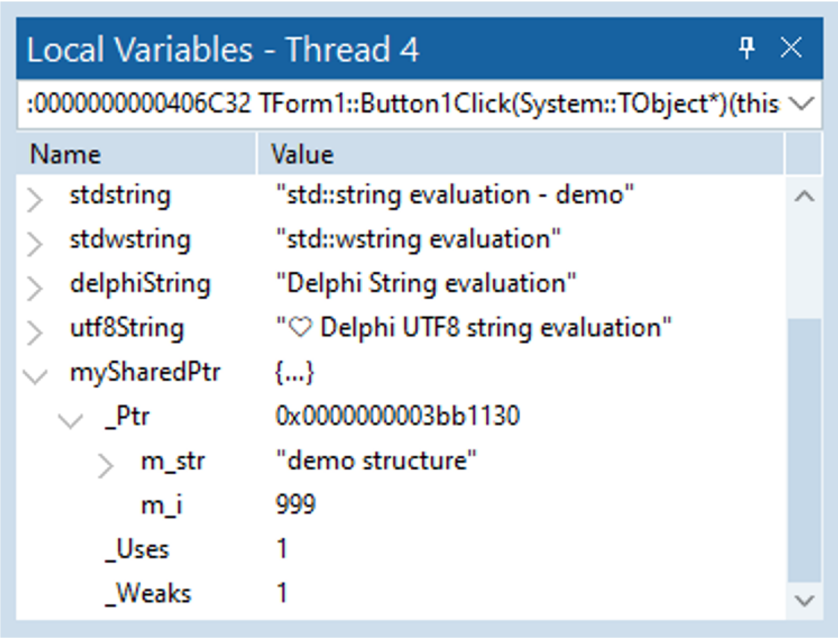
New C++ Builder 10.4 Features

New Windows 64-bit C++ Debugger

New modern LLDB based debugger for Windows

- Address key C++ debugging scenarios, stability, and features
- Better evaluation and inspection
- Includes ‘formatters’:
  - Allows you to evaluate and inspect complex types, including STL types (vector, dequeue, map etc), strings, and even your own custom complex types
- New foundation for future C++ debugger extensions

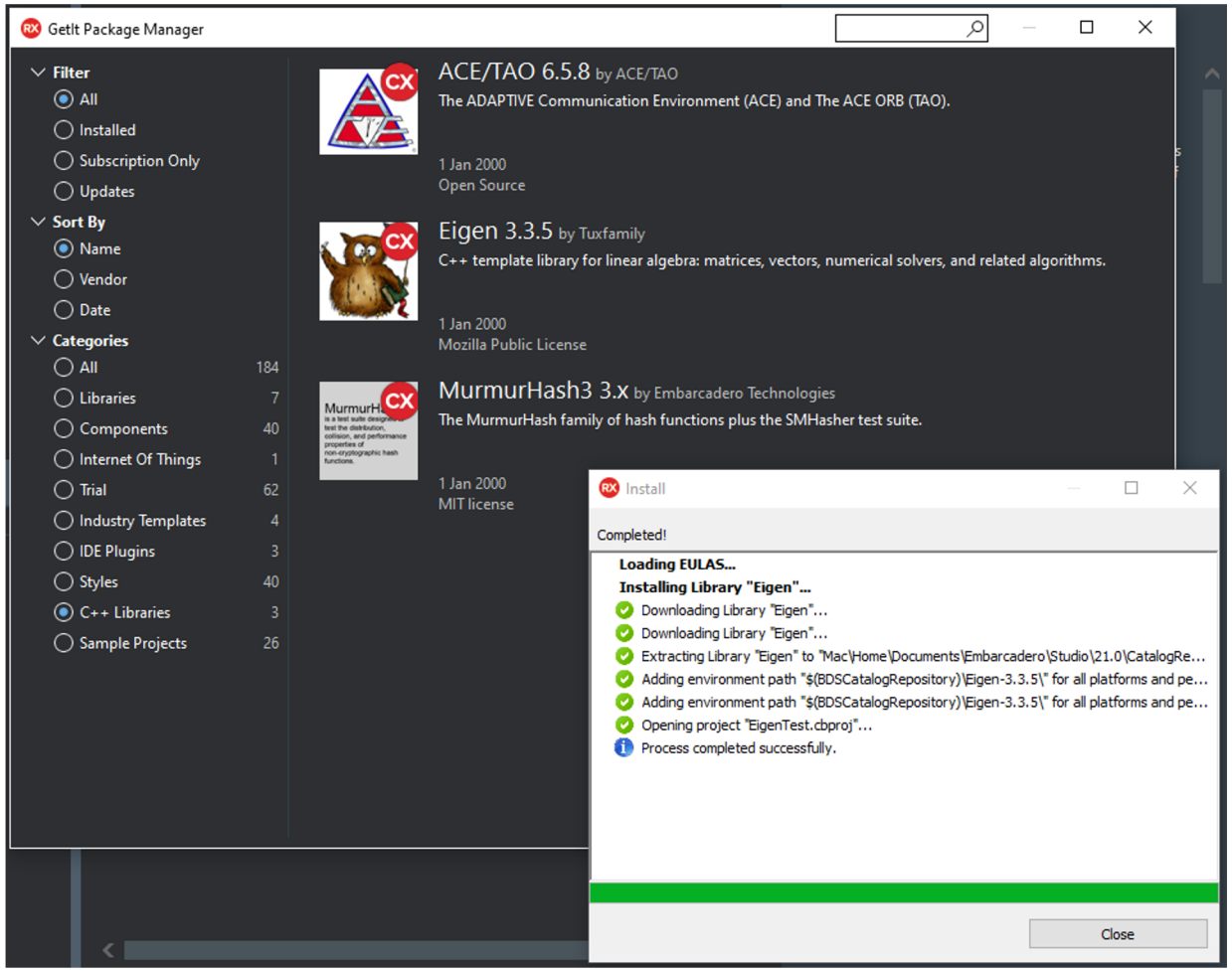




Expanded C++ Libraries Support

Support for common useful open source C++ libraries

- Several new key libraries can be downloaded via the GetIt Package Manager:
  - libSIMDpp
  - NemaTode
  - SDL
  - and others
- Boost
- CMake and RTL work to improve library compatibility



C++ – Other Enhancements

Toolchain performance and quality improvements

- Many improvements to CMake support
- Many C++ compiler enhancements, debug info generation, language extensions such as properties and closures, and exception handling
- The RTL has been updated with several missing methods that may be relied on by third party or open source C++ libraries
- General improvements, e.g. a new variadic version of Format(), and two new FreeAndNil signatures
- Support for ARC removal on mobile.

Questions?

1. Components (VCL / FMX)
2. Common Libraries (RTL)
3. Platform APIs (‘Touch the Metal’)

Any Questions?

embarcadero®