README

# Building and Running the LLVM Libcxx C++ Test Suite for RTX64

This README walks you through the steps needed to build and run the LLVM Libcxx C++ test suite for RTX64 available in GitHub at https://github.com/IntervalZero/llvm-project/tree/LLVM-RTX64.

**NOTE:** Before you begin, make sure you have downloaded all available files from the above location.

## Prepare the Test Suite Directory

1. Open Explorer and navigate to the directory to which the GitHub files were downloaded.
2. Copy `RTX64_LibcxxTestSuite` to the `C:\` drive on the machine where you will run the test suite.
3. Copy the `\Libcxx\test` folder to the test suite directory.
4. Copy `assert.h` and `cassert.h` files into the test/support directory
5. Remove the `llvm` folder from the test suite directory, if it is present.
6. Make sure the test suite directory is not Read Only.

## Build and Run the Test Suite

1. Start Powershell as administrator.
2. Allow scripts for the machine by running this Powershell command:
   ```
   Set-ExecutionPolicy bypass -force
   ```
3. Build `/RTX64_LibcxxTestSuite/RTX64_LibcxxTestSuiteMonitor` in either the Debug x64 or Release x64 configuration.

4. Place the resulting EXE in the `C:\RTX64_LibcxxTestSuite` directory on the machine where the test suite will run.

5. Open `Template.vcxproj` file and navigate to the below mentioned line to change Windows SDK version

   ```
   <WindowsTargetPlatformVersion>10.0.22621.0</WindowsTargetPlatformVersion>
   ```

6. Open `TestSuite.props` file to set PlatformToolset, TEST_STD_VER, MSVC_TEST_STD_VER , and MSVC_PLATFORM components as per desire visual studio and C++ version

7. Open `SetUpTestSuite.ps1`

   a. Navigate to the below mentioned line to update visual studio path and Windows SDK version

   ```
   #brute force some environments to allow us to construct an SLN

   C:\"Program Files\Microsoft Visual
   Studio"\2022\Professional\VC\Auxiliary\Build\vcvarsall.bat x64 10.0.22621.0
   ```

   b. Navigate to the **$CPPVersion** variable to update C++ version

   c. Navigate to the **$VSPlatform** variable to update visual studio platform

8. Run `SetUpTestSuite.ps1`

9. Close Powershell.

10. Start Powershell again as administrator

11. In RTX64 Server Console, set **Log File** to "C:\CPPTestsLog.txt" and enable **Log output to File**.

12. Open `BuildAndRun.ps1`

    a. Navigate to the below mentioned line to update visual studio details, here update MSBuild location and visual studio platform version

    ```
    #Execute the test

    $buildOutput = C:\"Program Files\Microsoft Visual
    Studio\2022\Professional\MSBuild\Current\Bin\amd64\MSBuild.exe"
    $projects[$projectIndex]  '/t:build' $configString '/p:Platform=x64'
    '/p:PlatformToolset=v143' '/v:d'
    ```

    b. Navigate to the below mentioned line to update non-default RTSS processor (Script uses 8 as a non-default processor)

    ```
    C:"\Program files\IntervalZero\Rtx64\Bin\RtssRunAndWait.exe" /p 8
    $ProgramPath
    ```

13. Run `BuildAndRun.ps1`

    **NOTE: The `BuildAndRun` script uses Visual Studio 2022.**

14. This generates two CSVs: one for RtssDebug results, and one for RtssRelease results.

15. Do the following:

    a. Copy the contents of the generated RtssDebug CSV.

    b. Open `LibcxxTestSuiteResultsTemplate.xlsx` and paste the RtssDebug CSV content into the matrix.

    c. Highlight row 3.

    d. Press **Ctrl+Shift+Down** to highlight all tests.

    e. Press **Ctrl+C** to copy all tests.

    f. Paste the tests into the RtssDebug sheet of the Results Template Matrix.

    g. Copy the content of the generated RtssRelease CSV and paste it into the matrix.