

## SAPIENT Test Harness Source Code Build Note

---

This document details the process carried out to build and run the SAPIENT Test Harness source code as it is provided on Dstl's GitHub [SAPIENT Middleware and Test Harness repository](#).

The described process was completed using:

- SAPIENT Test Harness source code, version 2.7.4
- Windows 10 Pro, version 20H2, build 19042.1110
- Visual Studio 2019, version 16.10.4
- Microsoft .NET Framework, version 4.8.04084

Additionally, to run the compiled binaries, the following components are required:

- PostgreSQL, version 12.6.1
- pgAdmin 4, version 5.0

### Prerequisites

Prior to compiling any of the components of the SAPIENT Test Harness, the aforementioned dependencies must be installed. See [Installations](#) for installation settings and additional components.

### Incompatibilities

Version 2.7.4 of the SAPIENT Test Harness is incompatible with a default installation of **PostgreSQL 13**. For the SAPIENT Test Harness to initialise a database, create the associated tables and connect, PostgreSQL 12 required.

### Installations

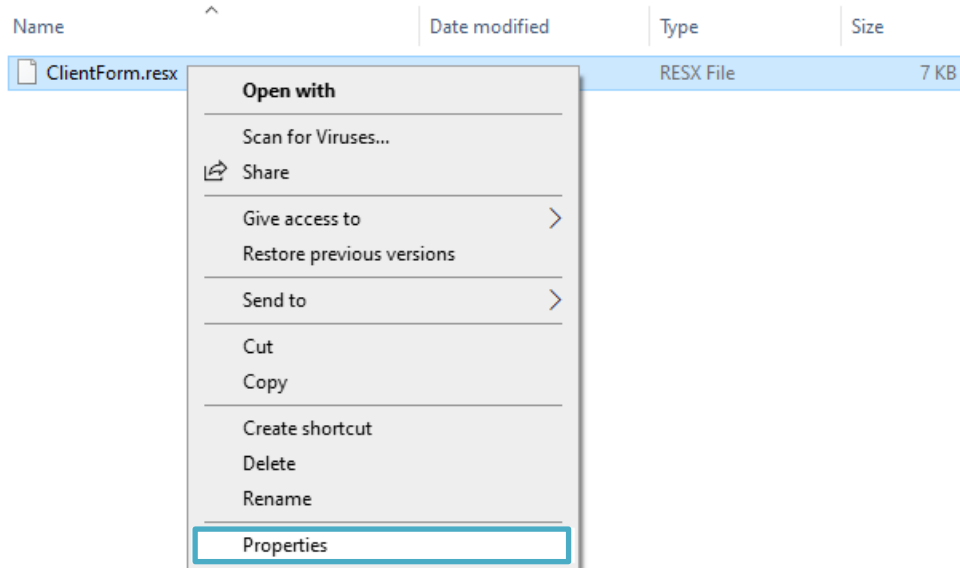
Windows 10 Pro was installed using the default installation settings, Visual Studio 2019 was installed with additional **C# and C#.NET components**, and the Microsoft .NET Framework was installed with default settings.

PostgreSQL was installed using the default installation settings, with the **“data” directory being saved outside the main “pgsql” directory**. pgAdmin was installed using the default installation settings.

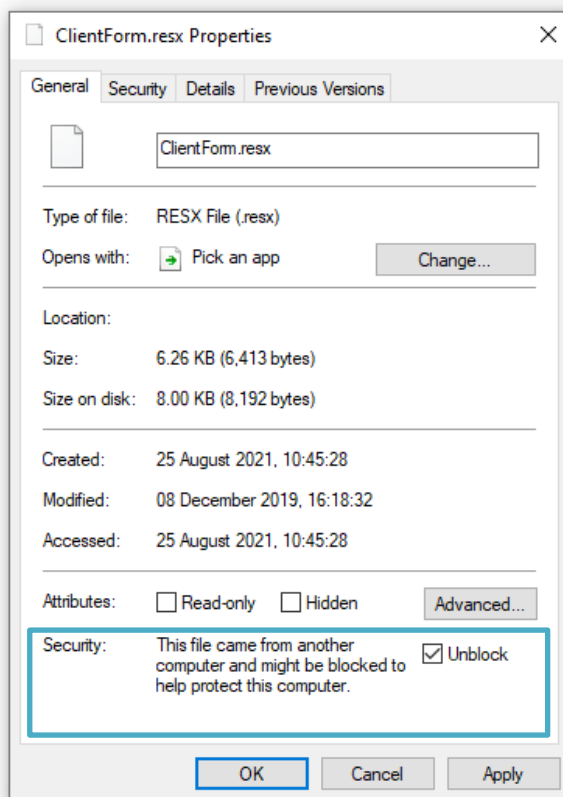
## Build Preparation

Before opening the Visual Studio Solution files (.sln), all NET Resource files (.resx) need to be unblocked. This can be done by:

1. **Right-clicking the .resx file** in Windows Explorer and selecting **Properties**.

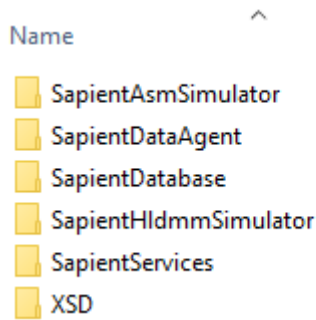


2. **Checking the “Unblock” box** at the bottom of the Properties dialog box.



3. **Clicking “OK”**.

4. Ensure the root directory of the SAPIENT Test Harness is organised as show below:

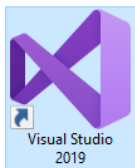


The SAPIENT Test Harness components will not compile without the SapienDatabase, SapienServices and XSD directories located as shown.

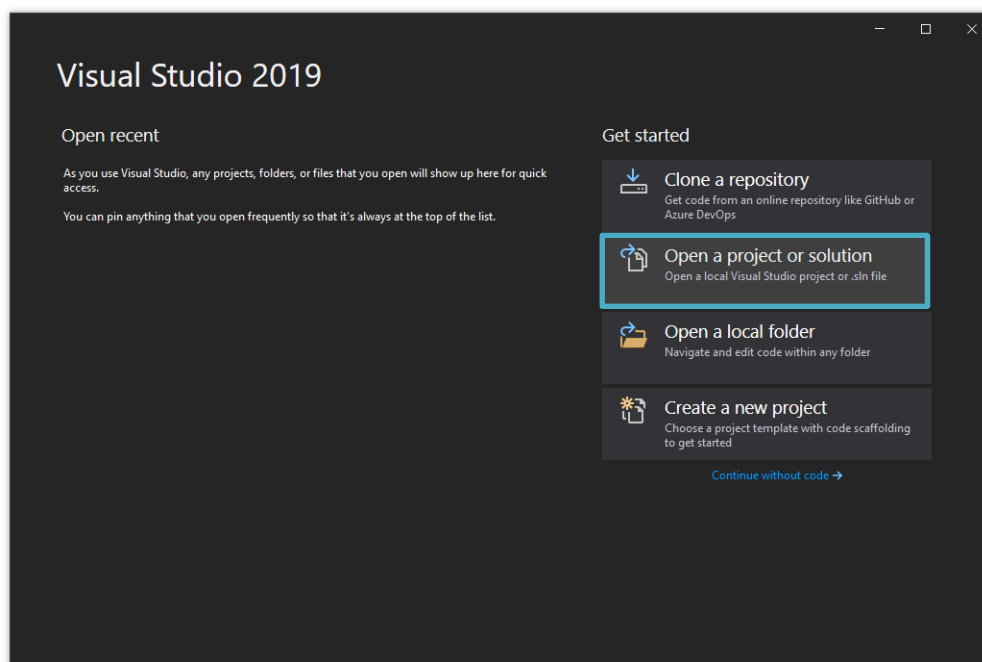
## Build

To build the components of the SAPIENT Test Harness, follow the instructions below, demonstrated with the SapienAsmSimulatorV3 project. These steps are repeatable for the SapienDataAgentV3 and SapienHldmmSimulatorV3 projects.

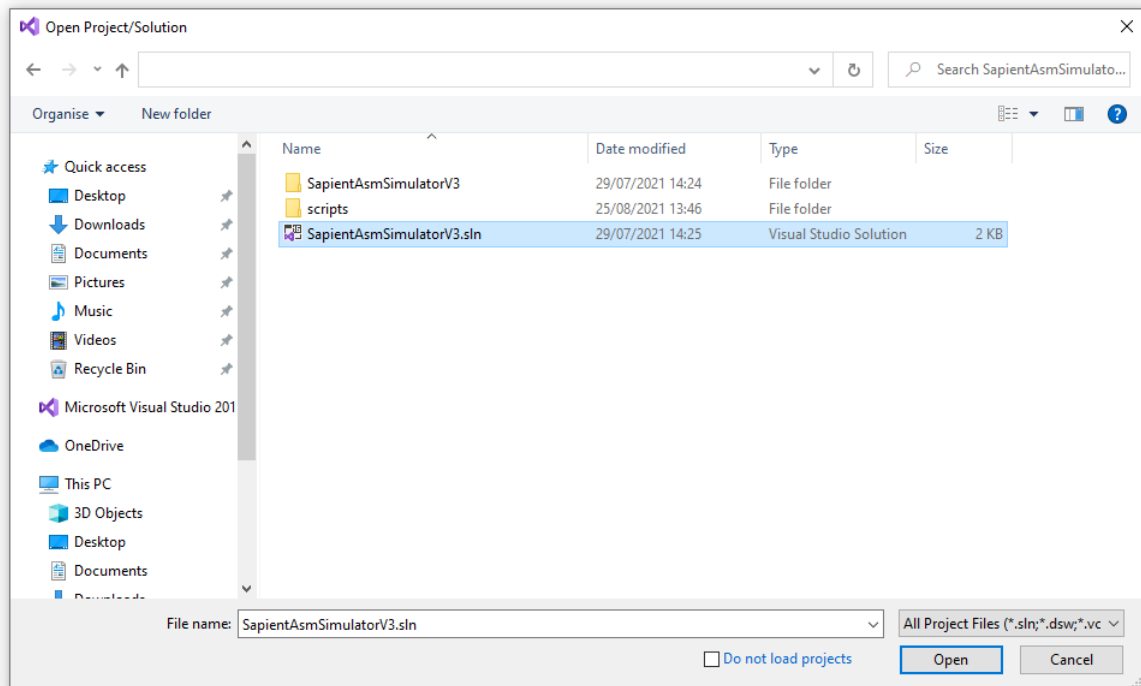
1. Open **Visual Studio**.



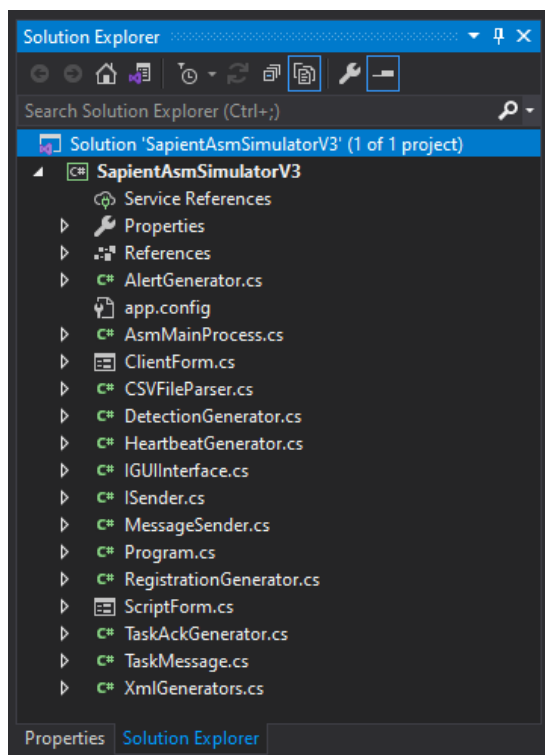
2. When Visual Studio loads, select **“Open a project or solution”**.



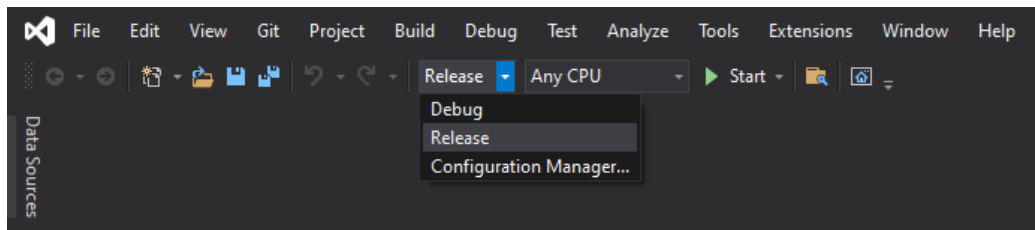
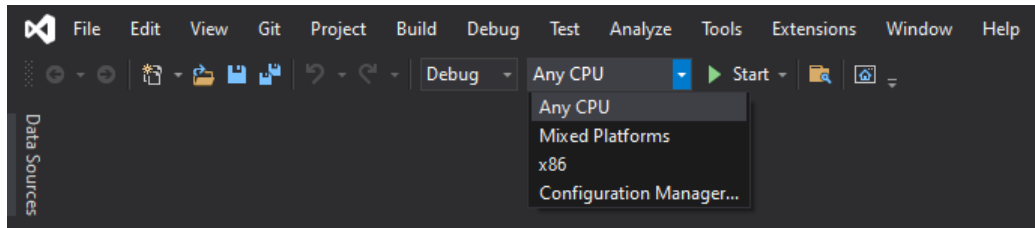
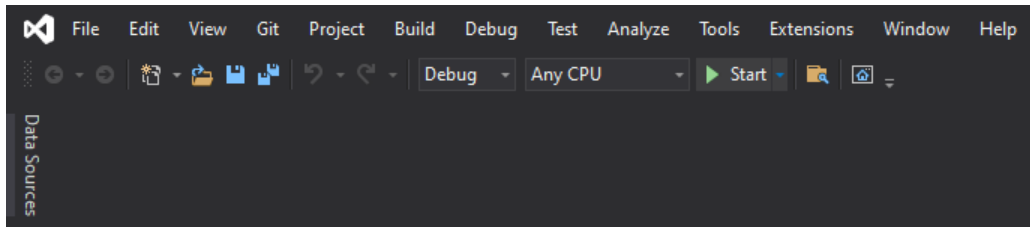
3. Navigate to the directory of the SAPIENT Test Harness source code and select the **Visual Studio Solution (.sln)** file. Click **“Open”**.



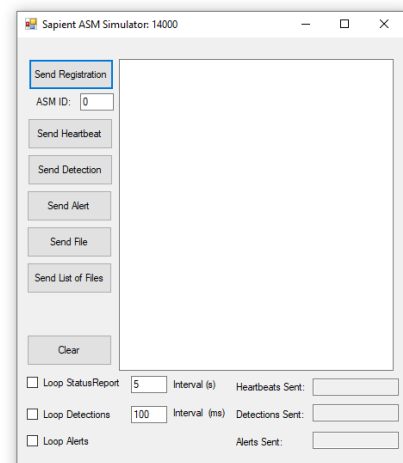
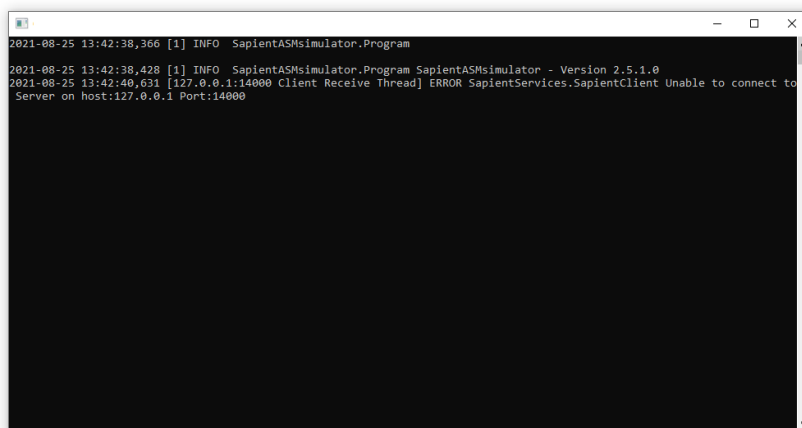
4. Having opened the Solution, all project files should be visible in the **Solution Explorer**. Depending on which component of the SAPIENT Test Harness is being built, this will vary.



5. Before building the project, configure the project to be built:

a. As a **Release**.b. For **Any CPU**.6. Click **“Start”**.


Having run through these steps, the Command Prompt window should open, shortly followed by the main window.



## Run

To start any components of the SAPIENT Test Harness, navigate to the project's main directory, then the **"scripts"** sub-directory. Double-click to run one of the **Windows Batch Files** named "Start[Component]1".

For example, to start **ASM1**:

 startASM1	12/03/2019 15:04	Windows Batch File	1 KB
---	------------------	--------------------	------

This script:

- Navigates to the project's **/bin/release/** directory.
- Starts the relevant executable with a specified integer parameter. This dictates the port that the socket connection runs on.

```
cd ../SapientASMSimulatorV3/bin/release
start "ASM1" SapientASMsimulator.exe 1
```

## Testing the Build

The following tests were carried out on the SAPIENT Test Harness, built as per the instructions above.

Action	Expected Response	Passed
Start SAPIENT Data Agent (SDA)	SDA shows ASM connection – User Interface shows "Clients Connected 1"	✓
Start HLDMM Data Agent (HDA)	SDA shows HDA connection – User Interface "Tasking Connected" checkbox is checked	
Start ASM Simulator (ASM)	HDA shows SDA connection – User Interface "Clients Connected 1"	
Start HLDMM Simulator (HLDMM)	HDA shows HLDMM connection – User Interface "Tasking Connected" checkbox is checked	
ASM, click "Send Registration" button	SDA shows receipt of registration message – "Sensor Registration" count displayed as "1"  HDA shows receipt of registration message – "Sensor Registration" count displayed as "1"	✓

	<p>ASM display window shows:</p> <ul style="list-style-type: none"> <li>• “Sensor Registration Sent”</li> <li>• “SensorRegistrationACK Received”</li> <li>• “ASM ID: 1”, “Latency(ms): [value]”</li> </ul> <p>Check message has appeared in the logs. If running with a database, verify Registration has been added to database “registration” table using pgAdmin</p>	
ASM, click “Send Heartbeat” button	<p>SDA shows receipt of StatusReport message – “Status Report” count displayed as “1”</p> <p>ASM display window shows:</p> <ul style="list-style-type: none"> <li>• “Heartbeats Sent: 1”</li> </ul> <p>Check message has appeared in the logs. If running with a database, verify Registration has been added to database “registration” table using pgAdmin</p>	✓
ASM, click “Send Detection” button	<p>SDA shows receipt of DetectionReport message – “Detection Report” count displayed as “1”</p> <p>ASM display window shows:</p> <ul style="list-style-type: none"> <li>• “Detections Sent: 1”</li> </ul> <p>Check message has appeared in the logs. If running with a database, verify DetectionReport has been added to the database using pgAdmin</p>	✓
HLDMM, set Sensor ID textbox to 1, click “Send Task” button	<p>HDA shows receipt of SensorTask message – “Sensor Task” count displayed as “1”</p> <p>SDA shows receipt of SensorTask message – “Sensor Task” count displayed as “1”</p> <p>ASM shows receipt of tasking message and sends SensorTaskACK to SDA</p>	✓

	<p>ASM display window shows:</p> <ul style="list-style-type: none"> <li>• “SensorTaskACK Sent”</li> <li>• “SensorTask Received: ASM ID 0 Latency(ms): [value]”</li> </ul> <p>SDA shows receipt of SensorTaskACK message – “Sensor Task Ack” count displayed as “1”</p> <p>HDA shows receipt of SensorTaskACK message – “Sensor Task Ack” count displayed as “1”</p> <p>HLDMM display window shows:</p> <pre>SensorTaskACK: &lt;?xml version="1.0"?&gt; &lt;SensorTaskACK&gt;   &lt;timestamp&gt;14/10/2014 14:08:41&lt;/timestamp&gt;   &lt;sensorID&gt;1&lt;/sensorID&gt;   &lt;TaskID&gt;0&lt;/TaskID&gt;   &lt;Status&gt;Accepted&lt;/Status&gt; &lt;/SensorTaskACK&gt; Latency (ms)</pre> <p>Check message has appeared in the logs. If running with a database, verify “SensorTask” has been added to the database using pgAdmin, verify “ack” field is “Accepted”</p>	
HLDMM, click “Send Detection” button	<p>HLDMM displays “Detection Sent”</p> <p>HDA “Detection” count displayed as “1”</p> <p>Check message has appeared in the logs. If running with a database, verify DetectionReport has been added to the database using pgAdmin</p>	✓
<p>HLDMM, set “Sensor ID” textbox to any integer value, except 1</p> <p>HLDMM, click “Send Task” button</p>	<p>HDA shows receipt of SensorTask message with invalid Sensor ID – “Error On ID” count displayed as “1”</p> <p>SDA unchanged</p> <p>ASM unchanged</p>	✓



	<p>HLDMM display windows shows (e.g. for Sensor ID 35):</p> <pre>SensorTaskACK: &lt;?xml version="1.0"?&gt; &lt;SensorTaskACK&gt;   &lt;timestamp&gt;14/10/2014 14:20:28&lt;/timestamp&gt;   &lt;sensorID&gt;35&lt;/sensorID&gt;   &lt;TaskID&gt;0&lt;/TaskID&gt;   &lt;Status&gt;Rejected&lt;/Status&gt;   &lt;Reason&gt;No ASM with this ID&lt;/Reason&gt; &lt;/SensorTaskACK&gt; Latency (ms)</pre> <p>Verify "SensorTask" has been added to the database using pgAdmin, verify "ack" field is "Rejected No ASM with this ID"</p>	
ASM, click "Send Alert" button	<p>SDA shows receipt of Alert message, count displayed as "1"</p> <p>ASM display window shows:</p> <ul style="list-style-type: none"> <li>• "Alerts Sent: 1"</li> </ul> <p>Check message has appeared in the logs. If running with a database, verify "SensorTask" has been added to the database using pgAdmin</p>	✓
<p>HLDMM, set Sensor ID textbox to "1"</p> <p>HLDMM, set Alert ID textbox to "0"</p> <p>HLDMM, click "Send Alert Response" button</p>	<p>HLDMM displays:</p> <pre>&lt;?xml version="1.0"?&gt; &lt;AlertResponse&gt;   &lt;timestamp&gt;14/10/2014 14:20:28&lt;/timestamp&gt;   &lt;sourceID&gt;1&lt;/sourceID&gt;   &lt;alertID&gt;0&lt;/alertID&gt;   &lt;status&gt;Acknowledge&lt;/status&gt;   &lt;reason&gt;Reason&lt;/reason&gt; &lt;/AlertResponse&gt;</pre> <p>HDA "Alert Response" count displayed as "1"</p> <p>SDA "Alert Response" count displayed as "1"</p> <p>ASM shows "AlertResponse Received alertID: 0"</p>	✓
Close SDA, close HDA, close ASM, close HLDMM	SDA closes, HDA closes, ASM closes, and HLDMM closes. Check applications not left hanging in Task Manager	✓

## Example Logs

To view example logs from the tests carried out in [Testing the Build](#), see the linked TXT files for the relevant SAPIENT Test Harness components:

<b>SDA</b>	<a href="https://github.com/dstl/SAPIENT-Middleware-and-Test-Harness/blob/main/Example/SDA1.txt">https://github.com/dstl/SAPIENT-Middleware-and-Test-Harness/blob/main/Example/SDA1.txt</a>
<b>HDA</b>	<a href="https://github.com/dstl/SAPIENT-Middleware-and-Test-Harness/blob/main/Example/HDA.txt">https://github.com/dstl/SAPIENT-Middleware-and-Test-Harness/blob/main/Example/HDA.txt</a>
<b>ASM</b>	<a href="https://github.com/dstl/SAPIENT-Middleware-and-Test-Harness/blob/main/Example/ASM1.txt">https://github.com/dstl/SAPIENT-Middleware-and-Test-Harness/blob/main/Example/ASM1.txt</a>
<b>HLDMM</b>	<a href="https://github.com/dstl/SAPIENT-Middleware-and-Test-Harness/blob/main/Example/HLDMM.txt">https://github.com/dstl/SAPIENT-Middleware-and-Test-Harness/blob/main/Example/HLDMM.txt</a>