SAPIENT BSI Flex 335 v2 Test Harness – User Manual

Dstl, Cyber and Information Systems

The Science Inside

Report Reference	Date of Issue	Principal Author	Lead Technical Reviewer	Project Manager
DSTL/PUB161312 v1	31/07/2024	Colley Stuart R	Bill Shay	Arthur Stratford

Introduction

The SAPIENT Test Harness software is a software tool to enable developers of SAPIENT components (edge nodes, fusion nodes, or middleware applications) to test the compliance of their component with the SAPIENT Standard – BSI Flex 335. The software would typically be used during the development and testing stage of a SAPIENT component to ensure compliance of the component with the standard.

The manual is for version 5.2.3 of the Test Harness software which is only compliant with version 2 of BSI Flex 335. The software is not capable of testing compliance with earlier versions of the SAPIENT standard.

It should be noted that successful testing of a component with the test harness does not negate the need for testing a SAPIENT component within a real, live SAPIENT system. Whilst the Test Harness can test compliance of the SAPIENT messages to the standard, live system testing is more likely to identify any performance issues that may need addressing, and ensure the 'practicalities' that make a SAPIENT system work, e.g. good time synchronisation, and low latencies are suitably catered for.

This manual covers the following aspects of the SAPIENT Test Harness:

1. How to build the software from the source code
2. Installation of the software
3. Configuration of the software
4. Running the software

Finally, it should be noted that the test harness software has a long history dating back to early versions of the SAPIENT standard nearly ten years ago. Updates to the test harness have occurred on a somewhat ad-hoc basis over the years, and some of the terminology used to describe parts of a SAPIENT system has also changed in this time, but this has not been reflected in the terminology used in the test harness.

BSI Flex 335 v2 uses the term 'Edge Node' to refer to a sensor or effector; in the test harness this is referred to as an ASM (Autonomous Sensor Module). The term 'Fusion Node' in BSI Flex 335 v2 is referred to as a DMM (Decision Making Module) or HLDMM (High Level Decision Making Module) in the test harness software.

Building the software from source code and Installation of the software

Two configuration of target system have been tested to build the test harness from the source code, listed below in Table 1. Other system configurations may also successfully build the software from the source code, but users will need to conduct appropriate testing on alternative configurations to satisfy themselves this is the case.

Table 1: Build Test Configurations

Prerequisites	Configuration 1	Configuration 2
Microsoft Visual Studio (x64)	Visual Studio Professional 2022 LTSC 17.6.2	Visual Studio Community 2022 17.9.6
Microsoft .NET 6 SDK (x64)	6.0.408	6.0.421
Microsoft Windows (x64)	Windows 10 Business Edition 22h2	Windows 11 Home Edition 23H2
PostgresSQL (x64)	12.14.2	12.18.1

Users should ensure suitable versions of Microsoft Visual Studio 2022 and the Microsoft .NET 6 SDK (Software Development Kit) a pre-installed before commencing to build the Test Harness; refer to Table 1 for suitable versions of these packages.

The following section describes the installation and configuration of the required third-party NuGet packages within the software development environment.

Table 2: Installation of NuGet packages

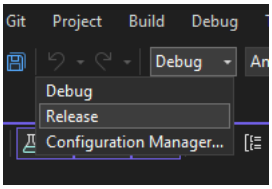
Description	The following describes the steps to copy the NuGet packages to the development environment and configure Microsoft Visual Studio with the location.		
Prerequisites	A software development environment containing an installation of Microsoft Visual Studio 2022 The prerequisite NuGet packages downloaded from www.nuget.org		
Step	**Operation**	**Value**	**Notes**
1.	Copy "*.nupkg".	Copy all the NuGet packages to "C:\SAPIENT\nupkg"	Relevant NuGet packages are listed in Software Packages, Versions, and Licences section of this document.
2.	Start "Visual Studio 2022".		Start Microsoft Visual Studio to configure the NuGet package location.

3.	Click "Continue without code".		Start Microsoft Visual Studio without loading a project or solution file.
4.	Choose "Package Manager Settings".	Tools > NuGet Package Manager > Package Manager Settings	From the main menu open the settings for the NuGet Package Manager.
5.	Choose "Package Source".	NuGet Package Manager > Package Source	From the navigation menu choose the NuGet pack source option.
6.	Click "+".		Click the plus button to add a new package source location.
7.	Enter "Source".	C:\SAPIENT\nupkg	Enter the new NuGet package source into the edit box.
8.	Click "Update".		Click update to confirm the new NuGet package location.
9.	Click "OK".		
10.	Close "Visual Studio 2022".		
Outcome	The NuGet packages required to build the SAPIENT Test Harness have been installed within the development environment.		

The following section describes using Microsoft Visual Studio to build the SAPIENT Test Harness from the source code.

Table 3: Build SAPIENT Test Harness

Description	The following describes the steps to load and building the SAPIENT Test Harness source code using Microsoft Visual Studio.		
Prerequisites	The SAPIENT Test Harness source code from GitHub.		
Step	**Operation**	**Value**	**Notes**

1.	Copy "*".	Copy all the source code to "C:\SAPIENT\workspace\sapienttestharness"	Copy the source code files and folders structure to the specified folder within the development environment.
2.	Start "Visual Studio 2022".		Start Microsoft Visual Studio.
3.	Click "Open a project or solution".		Click the open project or solution button. This will open a dialog titled "Open Project/Solution".
4.	Browse to "SapientTestHarness.sln".	"C:\SAPIENT\workspace\sapienttestharness\SapientTestHarness.sln"	Browse to the location of the SAPIENT Test Harness solution.
5.	Click "Open".		Open the SAPIENT Test Harness solution.
6.	Select "Release"		On the toolbar, choose Release from the Solution Configurations list
7.	Choose "Build Solution".	Build > Build Solution	Build the SAPIENT Test Harness and wait for the build process to complete.
8.	Close "Visual Studio Professional 2022".		Once the build process is complete close Microsoft Visual Studio.
Outcome	A set of SAPIENT Test Harness executables built from the source code.		

The following describes the process of copying and configuring the SAPIENT Test Harness executables so that they are in a structure suitable to be hosted on the Self-host virtual machine template.

Table 4: Configure and copy executables

Description	The following steps copy the SAPIENT Test Harness Executables into the "SAPIENT Test Harness BSI Flex 335 v2" folder and set the executables configuration files.		
Prerequisites	A set of SAPIENT Test Harness executables built from the source code.		
Step	**Operation**	**Value**	**Notes**
1.	Browse to "BuildExecutableFolder"	C:\workspace\sapienttestharness\BuildExecutableFolder	Browse to the folder containing the configuration batch file.
2.	Run "CopyExecutables.bat"		Run the copy executables batch file.
3.	Copy executables to CM	Copy files from C:\workspace\sapienttestharness\SAPIENT Test Harness BSI Flex 335 v2	Place the configured executables and ancillary files into a folder, ready for use.
Outcome	A set of configured SAPIENT Test Harness Executables and configuration and batch files are now located in the 'SAPIENT Test Harness BSI Flex 335 v2' folder, ready to be copied to the location where the User wishes to run the software from.		

Installation

The SAPIENT Test Harness software (the root folder) does not need to be installed in a particular location in order for the software to run correctly. If the software has been built from source, the 'C:\workspace\sapienttestharness\SAPIENT Test Harness BSI Flex 335 v2' folder should be copied to the desired location from which the software will be run. If extracting the binaries from a zip archive, this can be done in the desired location from which the software will be run.

Configuration of the Test Harness

The default configuration of the test harness when initially installed is for all of the test harness components to be located and run on one computer. The test harness is configured to 'self-test', i.e. the ASM Simulator component is configured to talk to, and test, the DMM Simulator component, and vice versa. In a real test scenario one of the components, either the ASM or DMM would be replaced with the component to be tested, and this component would most likely be located on hardware external to the test harness software. Thus to test an external SAPIENT component with the test harness requires a change of the test harness configuration files.

Configuration of the SAPIENT Test Harness components is split across multiple files, for the novice user it can be easy to forget which files have been modified, and which ones haven't. This can sometimes cause the test harness not to function as expected, thus it is recommended to

always keep a 'clean' copy of the software to hand, and to reconfigure the software from a clean installation in the event of any issues with the software.

There are a number of components to the SAPIENT Test Harness these are:

1. Simulated ASMs. It is possible for the Test Harness to run more than one simulated ASM at any time (three come pre-configured by default).
2. Simulated DMM. The functionality of the simulated DMM is much reduced compared to what would be expected from a 'real' DMM.
3. For each ASM there is a SDA (Sensor Data Agent). The SDA is the interface between the database in which messages are logged, and the ASM.
4. A HDA (HLDMM Data Agent). The HDA is the interface between the database in which all messages are logged, and the DMM
5. A database in which all messages from all SAPIENT components (ASMs and DMM) are logged.

A test scenario would normally consist of the component being tested, e.g. a sensor replacing a simulated ASM and being connected to the simulated DMM; or a real DMM being tested with the simulated ASMs from the test harness.

The test harness components are normally be started from their own 'batch file', e.g. 'Start ASM 1.bat', this file starts a Data Agent (SDA or HDA) and the simulated ASM or DMM. For ASMs, the batch file is capable of configuring the network port that the SDA and ASM simulator communicate on, and the SAPIENT 'node_id' of the ASM;

The Data Agents and simulated components (ASMs or DMM) each have their own folder containing the executables, another configuration file, and a log file. It should be noted that the port number and node_id specified in the batch file take precedence over the values specified in the configuration files within the folder for each component – thus if it is necessary to change the port number or node_id, it is recommended to do this in the batch file rather than the configuration file ('SapientASMsimulator.dll.config' for ASMs, or, 'SapientDmmSimulator.dll.config' for the DMM).

To test a SAPIENT ASM, one of the test harness Data Agents needs to be configured to communicate with (external) component. The 'SapientDataAgent.dll.config' file in the data agent folder needs to be edited to:

- Set the 'value' of the 'ClientAddress' to the IP address of the component being tested
- Set the 'value' of the 'DACommuniationPort' to the network port the ASM and Data Agent will communicate on – if you do not wish to use the default value.

The component being tested will need to be configured to connect to the IP address of the test harness and the network port specified by 'DACommunicationPort' in the Data Agent configuration file.

For a component being tested, it should have its own node_id which is a UUID v4 (Universally Unique IDentifier) which can be generated using an online service. It is recommended that the component being tested communicates with the Data Agent on the pre-configured network port, if possible, noting that a real SAPIENT system may use different networks ports.

Running the Test Harness

The Test harness can be started by running the appropriate batch files, e.g. 'Start DMM.bat' and/or 'Start ASM 1.bat'. The DMM should be started first since for some messages, notably the Registration Message, it needs to be ready to respond with an acknowledgment. The batch file

starts a Data Agent and a simulated component, if the simulated component is not needed because a 'real' component will be replacing it, then the simulated component can be closed leaving just the Data Agent ready to connect to the real component. Alternatively, the Data Agent can be run directly from the executable file (SapientDataAgent.exe) in the data agent folder.

The Test Harness Graphical User Interface (GUI) is not capable of setting every possible field that could be reported in a SAPIENT message, instead, within each simulated components folder, there are template SAPIENT messages (in JSON format) that can be customised to the Users needs, e.g. 'Default.Registration.json' for the Registration Message. The test harness software updates certain fields in the template messages when it sends them, notably the timestamp, and the node_id.

The first time the test harness software is run, it is not unusual to be shown a pop-up message from the Windows Firewall notifying the user that the test harness software is accessing the network. Users should select "Allow" to ensure the test harness functions correctly.

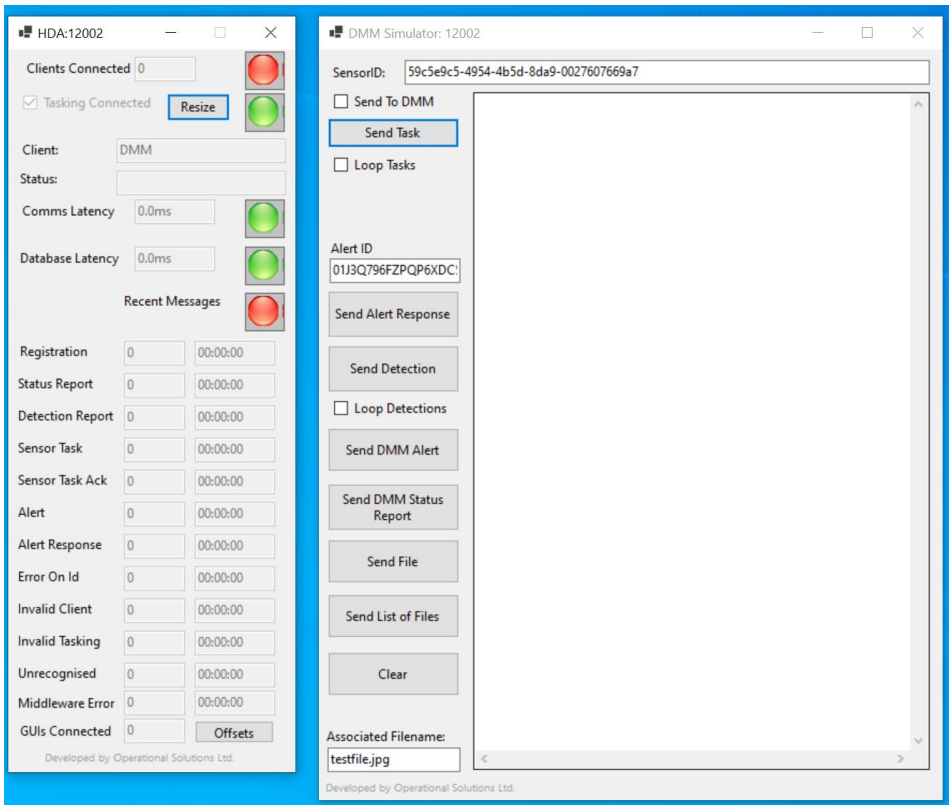


Figure 1: The HDA (DMM Data Agent) (left), and DMM Simulator (right)

The HDA and DMM simulator interfaces are shown above in Figure 1. The layout and display of the Data Agents is mostly self-explanatory. The HDA (DMM Data Agent) has a number of RAG (Red, Amber, Green) lights to highlight to the User any key system performance issues that may need investigation, notably, that clients are connected, latencies are within acceptable limits, and that messages are being received. The bottom half of the display summarises the number of valid (and invalid) messages of each type that have been received, and the time of the last message.

The functions of the test harness DMM simulator component (Figure 1, right hand side) are summarised in Table 5.

Table 5: DMM Simulator functions

Function	Description
Sensor ID	This is the (UUID v4) node_id of the ASM to which Task Messages and Alert Acknowledgement messages will go.
Send Task	This opens a dialog allowing the user to configure and send a single Task Message to the ASM specified by the Sensor ID, via the HDA.
Alert ID	This edit box allows the user to specify an ID (ULID) for an Alert Acknowledgement Message
Send Alert Response	This sends a single Alert Response to the node specified by the ASM ID via the HDA.
Send Detection	This sends a Detection Report to the HDA.
Send DMM Alert	This sends an Alert Message from the DMM to the HDA.
Send DMM Status Report	Sends a Status Message from the DMM to the HDA.
Send File	This opens a dialog allowing the user to select a file containing test data to be sent to the HDA.
Send List of Files	This opens a dialog allowing the user to select a text file containing a list of files from the same folder for sending in sequence. The user can control the sending of the files one-by-one.
Clear	This clears the text display on the right hand side of the user interface.

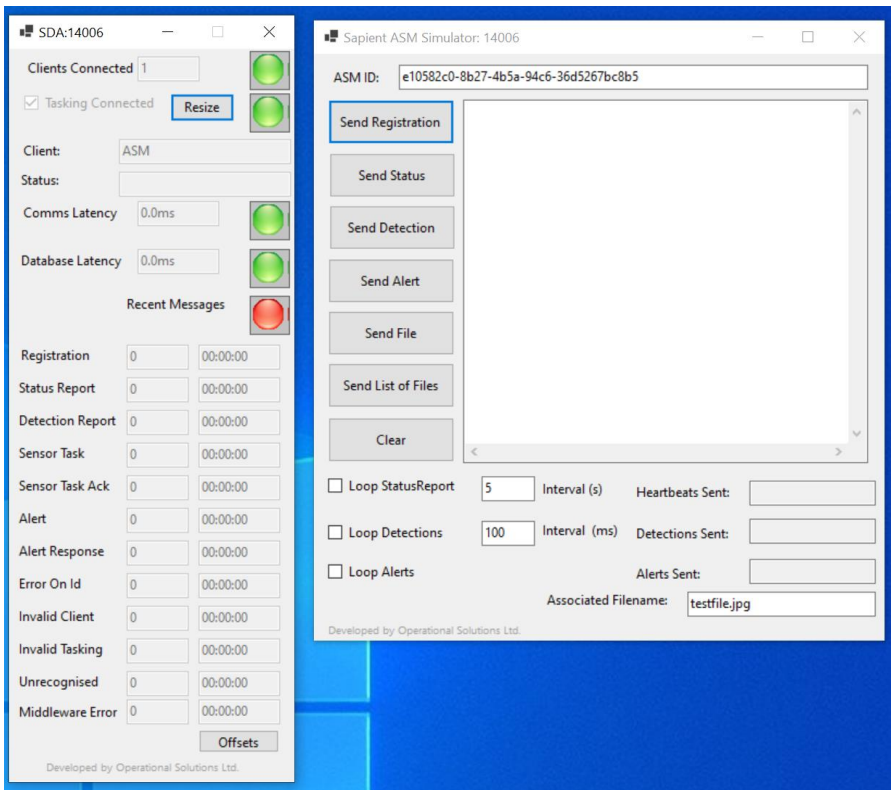


Figure 2: SDA (left), and ASM Simulator (right) interfaces

The SDA and ASM Simulator interfaces are shown above in Figure 2. The Sensor Data Agent (SDA) is identical to the HDA (described above) for all practical purposes. The functions of the ASM Simulator component (right hand side of Figure 2) are summarised below in Table 6.

Table 6: ASM Simulator functions

Function	Description
ASM ID	This is the UUID v4 of the node, assigned in either the configuration file, or the batch file used to start the test harness. When a message is sent, the node_id field of the message is updated with the ASM ID value displayed here.
Send Registration	This sends a Registration Message to the SDA. If the test harness is configured correctly, a Registration Acknowledgement message will be received, and the round-trip latency displayed.
Send Status	This sends a Status Message to the SDA. If the 'Loop StatusReport' box is checked, Status Reports will be automatically sent at the specified interval. If a Registration Message has not yet been acknowledged, a 'No ASM with this ID' error message will be displayed.
Send Detection	This sends a Detection Message to the SDA. If the 'Loop Detections' box is checked, Detection Messages will be automatically sent at the specified interval.
Send Alert	This sends an Alert Message to the SDA. If the 'Loop Alerts' box is checked, Alert messages will be automatically sent at the interval specified for looping detections.
Send File	This opens a dialog box allowing the user to select a file to send to the SDA.
Send List of Files	This opens a dialog box allowing the user to select a text file containing a list of files (from the same folder) that can be sent to the SDA in sequence. The user can control the sending of files one-by-one.
Clear	Clears the text display window on the right of the User Interface.

Software Packages, Versions, and Licences

The software packages used to build the software from source in Configuration 1, are listed below in Table 7.

Table 7: Software packages used to build the software with Configuration 1

Software	Version	Licence
castle.core.5.1.0.nupkg	5.1.0	Apache License 2.0
castle.core.5.1.1.nupkg	5.1.1	Apache License 2.0
fluentvalidation.11.1.0.nupkg	11.1.0	Apache License 2.0
fluentvalidation.11.5.1.nupkg	11.5.1	Apache License 2.0
fluentvalidation.aspnetcore.11.1.2.nupkg	11.1.2	Apache License 2.0
fluentvalidation.aspnetcore.11.3.0.nupkg	11.3.0	Apache License 2.0
fluentvalidation.dependencyinjectionextensions.11.1.0.nupkg	11.1.0	Apache License 2.0
fluentvalidation.dependencyinjectionextensions.11.5.1.nupkg	11.5.1	Apache License 2.0
google.protobuf.3.21.3.nupkg	3.21.3	BSD-3-Clause License
google.protobuf.3.21.4.nupkg	3.21.4	BSD-3-Clause License
google.protobuf.3.21.5.nupkg	3.21.5	BSD-3-Clause License

google.protobuf.3.24.3.nupkg	3.24.3	BSD-3-Clause License
google.protobuf.3.25.1.nupkg	3.25.1	BSD-3-Clause License
grpc.tools.2.47.0.nupkg	2.47.0	Apache License 2.0
grpc.tools.2.58.0.nupkg	2.58.0	Apache License 2.0
grpc.tools.2.60.0.nupkg	2.60.0	Apache License 2.0
log4net.2.0.14.nupkg	2.0.14	Apache License 2.0
log4net.2.0.15.nupkg	2.0.15	Apache License 2.0
microsoft.codecoverage.17.3.2.nupkg	17.3.2	Microsoft.CodeCoverage Microsoft Software License
microsoft.extensions.dependencyinjection.abstractions.2.1.0.nupkg	2.1.0	Apache License 2.0
microsoft.net.test.sdk.17.3.2.nupkg	17.3.2	Microsoft.NET.Test.Sdk Microsoft Software License
microsoft.testplatform.objectmodel.17.3.2.nupkg	17.3.2	Microsoft.TestPlatform.ObjectModel Microsoft Software License
microsoft.testplatform.testhost.17.3.2.nupkg	17.3.2	Microsoft.TestPlatform.TestHost Microsoft Software License
moq.4.18.2.nupkg	4.18.2	BSD-3-Clause License
newtonsoft.json.13.0.1.nupkg	13.0.1	MIT License
npgsql.6.0.5.nupkg	6.0.5	PostgreSQL License
NuGet.frameworks.5.11.0.nupkg[1]	5.11.0	Apache License 2.0
nunit.3.13.3.nupkg	3.13.3	NUnit License
nunit.analyzers.3.5.0.nupkg	3.5.0	NUnit.Analyzers License
nunit3testadapter.4.2.1.nupkg	4.2.1	MIT License
stylecop.analyzers.1.1.118.nupkg	1.1.118	Apache License 2.0
system.configuration.configurationmanager.7.0.0.nupkg	7.0.0	MIT License
system.diagnostics.eventlog.7.0.0.nupkg	7.0.0	MIT License
system.drawing.common.7.0.0.nupkg	7.0.0	MIT License
system.runtime.compilerservices.unsafe.6.0.0.nupkg	6.0.0	MIT License
system.security.cryptography.protecteddata.7.0.0.nupkg	7.0.0	MIT License
system.security.permissions.7.0.0.nupkg	7.0.0	MIT License
system.windows.extensions.7.0.0.nupkg	7.0.0	MIT License
ulid.1.2.6.nupkg	1.2.6	MIT License
.NET 6.0 SDK	6.0.408 win-x64	Windows SDK License
		.NET Library License
		Visual Studio 2019 License
		MIT License
SAPIENT Test Harness Source Code	5.2.3	Owned by DSTL
Microsoft Visual Studio Professional 2022 (x64)	17.6.2	Microsoft License Terms

[1] Version 5.11.0 of the nuget.frameworks package was unavailable when attempting to build the software to Configuration 2. Version 5.10.0 was used for building the software from source with Configuration 2.

Release conditions	Copyright
This document has been prepared for MOD and, unless indicated, may be used and circulated in accordance with the conditions of the Order under which it was supplied. It may not be used or copied for any non-Governmental or commercial purpose without the written agreement of Dstl. Parts of the document, which are identified, constitute valuable technical information controlled by Dstl or may be commercially sensitive in relation to third parties. *(Delete if not required)* Parts of the document have derived from separate funding and are being provided by way of information only. Such parts, which are identified, may not be released outside United Kingdom Government Departments without Dstl consent. *(Delete if not required)*	 Defence Science and Technology Laboratory UK Approval for wider use or release must be sought from: Intellectual Property Department Defence Science and Technology Laboratory Porton Down, Salisbury, Wiltshire SP4 0JQ

Report documentation page v5.0

* Denotes a mandatory field

1a.	**Report number: *** DSTL/PUB161312		**1b.**	**Version number:** v1
2.	**Date of publication: *** 31/07/2024		**3.**	**Number of pages:** 13
4a.	**Report UK protective marking: ***	UK OFFICIAL		
4b.	**Report national caveats: ***	NONE		
4c.	**Report descriptors: ***	NONE		
5a.	**Title: *** SAPIENT BSI Flex 335 v2 Test Harness – User Manual			
5b.	**Title UK protective marking: ***	UK OFFICIAL		
5c.	**Title national caveats: ***	NONE		
5d.	**Title descriptor: ***	NONE		
6.	**Authors: *** Colley Stuart R			
7a.	**Abstract: *** The SAPIENT Test Harness is a software tool that enables developers to test compliance of their SAPIENT component with the SAPIENT standard – BSI Flex 335 v2. This User Manual describes how the Test Harness software can built from its source code, installed, configured and run.			
7b.	**Abstract UK protective marking: ***	UK OFFICIAL		
7c.	**Abstract national caveats: ***	NONE		
7d.	**Abstract descriptor: ***	NONE		
8.	**Keywords:**			

Please note: Unclassified, Restricted and Confidential markings can only be used where the report is prepared on behalf of an international defence organisation and the appropriate prefix (e.g. NATO) included in the marking.

* Denotes a mandatory field

9.	**Name and address of publisher: *** Dstl Porton Down Salisbury Wiltshire SP4 0JQ	**10.**	**Name and address of funding source:**
11.	**Funding source contract:**		
12.	**Dstl project number:**	PJ100790	
13.	**Programme:**	Future Sensing Programme	
14.	**Other report numbers:**		
15a.	**Contract start date:**	**15b.**	**Contract end date:**
16.	**IP conditions for report: ***	MOD funded Dstl authored report	
17a.	**Patents:**	NO	
17b.	**Application number:**		
18.	**Release authority role:**		

Guidance on completing the report documentation page can be found on the Gov.UK website.