NORTH ATLANTIC TREATY ORGANIZATION

SCIENCE AND TECHNOLOGY ORGANIZATION

AC/323(MSG-118)TP/798

www.sto.nato.int

STO TECHNICAL REPORT **TR-MSG-118**

Development of Common Image Generator Interface (CIGI) V4.0 Compliancy Testing Tools Final Report

(Rapport final sur le développement des outils d'essai de conformité de la version 4.0 de l'interface commune de générateur d'image (CIGI))

Final Report of MSG-118.

Published July 2018

Distribution and Availability on Back Cover

NORTH ATLANTIC TREATY ORGANIZATION

SCIENCE AND TECHNOLOGY ORGANIZATION

AC/323(MSG-118)TP/798

www.sto.nato.int

STO TECHNICAL REPORT **TR-MSG-118**

Development of Common Image Generator Interface (CIGI) V4.0 Compliancy Testing Tools Final Report

(Rapport final sur le développement des outils d'essai de conformité de la version 4.0 de l'interface commune de générateur d'image (CIGI))

Final Report of MSG-118.

The NATO Science and Technology Organization

Science & Technology (S&T) in the NATO context is defined as the selective and rigorous generation and application of state-of-the-art, validated knowledge for defence and security purposes. S&T activities embrace scientific research, technology development, transition, application and field-testing, experimentation and a range of related scientific activities that include systems engineering, operational research and analysis, synthesis, integration and validation of knowledge derived through the scientific method.

In NATO, S&T is addressed using different business models, namely a collaborative business model where NATO provides a forum where NATO Nations and partner Nations elect to use their national resources to define, conduct and promote cooperative research and information exchange, and secondly an in-house delivery business model where S&T activities are conducted in a NATO dedicated executive body, having its own personnel, capabilities and infrastructure.

The mission of the NATO Science & Technology Organization (STO) is to help position the Nations' and NATO's S&T investments as a strategic enabler of the knowledge and technology advantage for the defence and security posture of NATO Nations and partner Nations, by conducting and promoting S&T activities that augment and leverage the capabilities and programmes of the Alliance, of the NATO Nations and the partner Nations, in support of NATO's objectives, and contributing to NATO's ability to enable and influence security and defence related capability development and threat mitigation in NATO Nations and partner Nations, in accordance with NATO policies.

The total spectrum of this collaborative effort is addressed by six Technical Panels who manage a wide range of scientific research activities, a Group specialising in modelling and simulation, plus a Committee dedicated to supporting the information management needs of the organization.

- AVT Applied Vehicle Technology Panel
- HFM Human Factors and Medicine Panel
- IST Information Systems Technology Panel
- NMSG NATO Modelling and Simulation Group
- SAS System Analysis and Studies Panel
- SCI Systems Concepts and Integration Panel
- SET Sensors and Electronics Technology Panel

These Panels and Group are the power-house of the collaborative model and are made up of national representatives as well as recognised world-class scientists, engineers and information specialists. In addition to providing critical technical oversight, they also provide a communication link to military users and other NATO bodies.

The scientific and technological work is carried out by Technical Teams, created under one or more of these eight bodies, for specific research activities which have a defined duration. These research activities can take a variety of forms, including Task Groups, Workshops, Symposia, Specialists' Meetings, Lecture Series and Technical Courses.

The content of this publication has been reproduced directly from material supplied by STO or the authors.

Published July 2018

ISBN 978-92-837-2123-9

Table of Contents

List of Figures and Tables

List of Acronyms

CCL	CIGI Class Library
CIGI	Common Image Generator Interface
COTS	Commercial Off The Shelf
CTT	Compliancy Testing Tools
GBR	Great Britain
GUI	Graphical User Interface
ICD	Interface Control Document
IEEE	Institute of Electrical and Electronics Engineers
IG	Image Generator
libCIGI	C++ CIGI interface library that supports V4.0
MOD	Ministry of Defence
MSG	Modelling and Simulation Group
NATO	North Atlantic Treaty Organisation
PDG	Product Development Group
PSG	Product Support Group
SISO	Simulation Interoperability Standards Organisation
STO	Science and Technology Organization
TAP	Technical Activity Proposal
TOR	Terms Of Reference
UDP	User Datagram Protocol
UK	United Kingdom
USA	United States of America

MSG-118 Membership List

CHAIR

Grant BAILEY
UK MoD
UNITED KINGDOM
Email: Grant.Bailey127@mod.gov.uk

MEMBERS

Marco DONARELLI
Selex ES
ITALY
Email: marco.donarelli@selex-es.com

Robbert KRIJNEN
TNO
NETHERLANDS
Email: robbert.krijnen@tno.nl

Paul KUIPER
National Aerospace Laboratory – NLR
NETHERLANDS
Email: Paul.Kuiper@nlr.nl

Gökçe YILDIRIM
Simsoft Bilgisayar Teknolojeri Ltd Sri
TURKEY
Email: gokce@simsoft.com.tr

CONTRIBUTORS

Patrick BAKKER
TNO
NETHERLANDS
Email: patrick.bakker@tno.nl

Roland HUMPHRIES
XPI Simulation Ltd.
UNITED KINGDOM
Email: roland.humphries@xpisimulation.com

Lewis LINN-COLE
XPI Simulation Ltd.
UNITED KINGDOM
Email: lewis.linn-cole@xpisimulation.com

Mark PETERSON
Binghamton Simulation Company
UNITED STATES
Email: mpeterson@bsc.com

Simon SKINNER
THALES UK Ltd.
UNITED KINGDOM
Email: simon.skinner@uk.thalesgroup.com

Neil SMITH
Dstl Portsdown West
UNITED KINGDOM
Email: nsmith@dstl.gov.uk

Musa SUMUN
XPI Simulation Ltd.
UNITED KINGDOM
Email: musa.sumun@xpisimulation.com

Arnoud VAN LEEUWEN
National Aerospace Laboratory – NLR
NETHERLANDS
Email: Arnoud.van.Leeuwen@nlr.nl

Adrian WEBB
Department of Defence
AUSTRALIA
Email: adrian.webb1@defence.gov.au

Development of Common Image Generator Interface (CIGI) V4.0 Compliancy Testing Tools Final Report

(STO-TR-MSG-118)

Executive Summary

MSG-118 was started to address the lack of compliance testing for the CIGI standard, compliance is considered very important to guarantee reusability of IG systems. Continuing the work started by the UK MOD funded Dstl led research project STACI into tools for CIGI compliance, the workgroup aimed to design and implement the tests needed to prove compliance against the standard.

The work was intended to cover the entire CIGI specification but CIGI itself is only planned to be tested against features that an IG claims to have. CIGI does not require a baseline set of features that every IG must implement. Because CIGI does not cover data formats the tests cannot cover the import / interpretation of models that have a significant effect on IG reusability.

The group started well with some very good discussions about the nature of the testing that should take place. Beyond this the work required a significant amount of technical analysis of the specification and further technical expertise to implement the tests. Because of the large cost of the initial approach the method was refined to reduce redundancy and streamline the process, but because of the volunteer nature of members participation there was not enough time available to complete the work.

The design of the compliance tools was completed and the tools are in a functional state although limited in terms of the tests and use cases. Further work would enable the test tools to cover further use cases and be of more general use to the CIGI community

The following recommendations are made:

1) Future groups developing technical products need suitably qualified technical staff and sufficient budget, planning and awareness of cost prior to group formation as this is critical to success; ideally budget needs should be identified in the exploratory team process with countries committing to provide the agreed share of resources.
2) Product development and maintenance needs to be considered by SISO and NMSG, executable products have users and issues beyond the end of the task groups that create them:
    - SISO and NMSG should have policies covering all kinds of product artifacts;
    - SISO and NMSG should have managed public repository areas (e.g. SourceForge, GitHub etc.) used to store and version technical content and product artifacts where these are evaluated as being suitable for public release to NATO industry and elsewhere; and
    - The use of standards within simulation and training have great benefit for interoperability and reuse, however the approach and cost of compliancy testing must be taken into account in business cases where use of standards are mandated.
3) Use virtual conferencing services that provide screen-share, transfer of control and video for regular task group meetings to save money and travel time.

Rapport final sur le développement des outils d'essai de conformité de la version 4.0 de l'interface commune de générateur d'image (CIGI)

(STO-TR-MSG-118)

Synthèse

Le MSG-118 a été créé pour pallier le manque d'essai de conformité de la norme CIGI, car la conformité est jugée très importante pour garantir la possibilité de réutiliser les systèmes de générateur d'image. Dans la continuité des travaux lancés par le projet de recherche STACI sur les outils de conformité CIGI, financé par le ministère de la Défense britannique et mené par le Dstl, le groupe de travail avait pour but de concevoir et réaliser les essais nécessaires pour démontrer la conformité à la norme.

Les travaux devaient couvrir toute la spécification CIGI, mais il n'est prévu de tester la CIGI elle-même qu'au regard des caractéristiques dont se réclame un générateur d'image. La CIGI n'a donc pas besoin d'un ensemble de caractéristiques de référence pour tous les générateurs d'image. Etant donné que la CIGI ne couvre pas les formats de données, les essais ne peuvent pas couvrir l'importation / interprétation de modèles qui ont un effet important sur la possibilité de réutiliser un générateur d'image.

Le groupe a bien démarré, avec de très bonnes discussions sur la nature des essais à organiser. En outre, ce travail a nécessité une analyse technique poussée et d'autres expertises techniques pour réaliser les essais. En raison du coût élevé de la démarche initiale, la méthode a été affinée dans le but de réduire les redondances et rationaliser le processus, mais à cause du principe de participation volontaire, le temps a manqué pour achever les travaux.

La conception des outils de conformité a été menée à terme et les outils sont fonctionnels, même s'ils ne s'appliquent qu'à un nombre limité d'essais et de cas d'utilisation. D'autres travaux permettraient de couvrir d'autres cas d'utilisation et seraient d'un usage plus général pour la communauté CIGI.

Les recommandations suivantes sont émises :

1) Les futurs groupes développant des produits techniques ont besoin de personnel technique qualifié et d'un budget suffisant, d'une planification et d'une sensibilisation aux coûts avant même la formation du groupe, car la réussite en dépend. Dans l'idéal, les besoins budgétaires devraient être identifiés pendant le processus de l'équipe exploratoire et les pays devraient s'engager à fournir la part de ressources convenue.

2) La SISO et le NMSG doivent prendre en considération les besoins en matière de développement des produits et de maintenance. Les produits exécutables ont des utilisateurs et présentent des problèmes après la fin des groupes de travail qui les ont créés :

   - La SISO et le NMSG devraient disposer de politiques couvrant tous les types de produits ;
   - La SISO et le NMSG devraient disposer d'espaces gérés dans des entrepôts de données (par exemple SourceForge ou GitHub) pour conserver le contenu technique et les produits et assurer le contrôle des versions. Dans ces espaces, le contenu et les produits seraient jugés adaptés ou non à une diffusion publique dans l'industrie de l'OTAN et ailleurs ; et

- L’utilisation de normes en simulation et en formation présente de grands avantages pour l’interopérabilité et la réutilisation, mais la démarche et le coût des essais de conformité doivent être pris en compte dans les analyses de rentabilisation lorsque l’utilisation des normes est obligatoire.

3) Il faudrait faire appel à des services de conférence virtuelle avec partage d’écran, transfert de contrôle et vidéo pour les réunions régulières du groupe de travail, afin d’économiser de l’argent et du temps dans les transports.

Chapter 1 − INTRODUCTION

This Report describes the work of the North Atlantic Treaty Organization (NATO) Modeling and Simulation (NMSG) Research Task Group MSG-118 "Common Image Generator Interface (CIGI) Compliancy Tools".

The Task Group was set up in 2012 with the purpose of creating a suite of tools for testing Image Generator (IG) compliance against the Simulation Interoperability Standards Organisation (SISO) CIGI standard V4.0.

Between 2012 and early 2016 the Task Group, comprising representatives from United Kingdom (lead nation), Netherlands, United States, Canada, Australia, Turkey and Italy, has performed a large amount of work to extract use cases, transform these use cases into test definitions, generate python test implementations and finally test the tests against available IGs. While the work is incomplete it is considered that completing the implementations would only require a modest directed budget.

The Report is broken down into several chapters, each describing a different aspect of the work:

- Chapter 2 – The task background, the need for compliance tools for CIGI and the starting state of the tools;
- Chapter 3 – The expected outputs;
- Chapter 4 – The programme of work undertaken and initial approach;
- Chapter 5 – The meetings and organisation of the group;
- Chapter 6 – Task group outputs, i.e. the current state of the compliance tools and tests;
- Chapter 7 – Outstanding issues and further work; and
- Chapter 8 – Conclusions and recommendations.

Chapter 2 − BACKGROUND

2.1 CIGI

The primary goal of the Common Image Generator Interface (CIGI) standard is to standardise the interface between a simulators host computer and the Image Generator (IG). The latest version (V4.0) of CIGI is the first open standard version, developed under the auspices of SISO. A standard interface improves interoperability and therefore helps reduce integration and through-life support costs.

The issue with high costs for integrating new or replacement IGs into simulators is exacerbated by the lack of an effective common interface standard. Efforts had begun to standardise the pre-existing Boeing owned CIGI 3.3. CIGI 3.3 and its previous versions had been used predominantly in the air domain, despite there is no other alternative and creation of a new standard would be unproven so standardisation of CIGI seemed the best option.

2.2 THE NEED FOR COMPLIANCE

It is important when a standard is created that there are also tools to check that applications using it are compliant with the standard. Many government procurement agencies (for example, USA, GBR) make use of simulation standards compulsory in contracts with industry suppliers to reduce the through life cost of procuring and maintaining simulator based training devices. CIGI is one of these standards.

Compliance ensures standard implementation which in turn ensures that systems are more future proof and interoperable, a standard without a compliance function is open to abuse / interpretation.

2.3 PREVIOUS WORK

Industry had not generated (common) tools for CIGI compliance since its initial release in 2001, UK MOD decided that to kick start the compliance effort it would fund the starting point for the tools required to test compliance. The tools were designed to cover as much of the compliance of CIGI as possible but given that much of CIGI is optional and the implementation / appearance of many features is left to the IG vendors' discretion this is hard to achieve.

The tools are comprised of a host analyser that provides a mechanism for both limited testing compliance of host applications to the standard and also as a debugging aid to host developers. The IG compliance tool provides a method for IG vendors to generate test profiles that can be run against their IG, some tests would be automated but many would require manual review, the tool will give a compliance report on completion.

2.4 MSG-118

Standards based acquisition is being led by government agencies and therefore the NMSG agreed that a government led research task group should be formed to ensure that nations had a major part in ensuring that standards development and testing against standards was robust and fair, with support from industry/ academia to provide some independence and wider availability to NATO nations and agencies. A co-operative approach was justified by the fact that the selected CIGI compliance process would be distributed as part of national capabilities.

In 2012, while work developing CIGI V4.0 was still progressing, the MSG-118 Task Group was formed, led by the United Kingdom (GBR) with the following nations taking an active role:

- Canada (CAN);
- Netherlands (NLD);
- Turkey (TUR); and
- United States (USA).

The purpose of the task group was to develop CIGI V4.0 Compliancy Testing Tools (CTT) and an associated testing process based on the work already completed by the UK MOD.

In August 2014 the CIGI V4.0 standard was released and both the MSG-118 Group and the CIGI Product Development Group (PDG) members made efforts to publicise this.

Chapter 3 – EXPECTED OUTPUTS

The activity description for MSG-118 [4] describes the main objectives for MSG-118 as being:

- *To investigate the implementation of the CIGI V4.0 Standard.*
- *To develop open source tools that can be utilised by NMSG nations to test Image Generators for CIGI V4.0 compliancy. This will include the production of a development plan including appropriate milestones.*

The MSG-118 Technical Activity Proposal / Terms of Reference (TAP/TOR) goes further, describing the objectives to be:

- *The CIGI Compliancy Testing Tools (CTT) plan will include requirements for manpower, hardware and software. In addition this development plan will describe how CIGI CTT work share will be distributed between participating nations.*
- *Ease of implementing, and a procedure for implementation, the CIGI standard alongside the need for compliancy tools. Also required will be compliancy criteria, compliancy tool requirements and the role of the user.*
- *As a living and open standard, CIGI is expected to evolve with time. The task group will therefore plan to allow for future upgrades and improvements to the testing tools.*
- *The plan will also describe how developed tools will be handed over to the SISO CIGI Product Support Group for future maintenance and support.*

The TAP/TOR also states expected deliverables as being:

- *Technical Report, other deliverable(s): Compliancy testing tools for the new SISO Open CIGI Standard and the associated testing process.*

From a more technical standpoint the team were expecting to deliver:

- *Complete compliance test suite that can be used to prove IG compliance to CIGI standard.*
- *IG tested via unit tests, automated where possible:*
  - *Each test targets a specific feature / function.*
  - *IG vendors select the tests that the IG should pass and discard the rest.*
  - *Tests should cover any standard functionality presented by the Interface Control Document (ICD) [1].*
  - *It should be possible for IG vendors to extend with additional tests for vendor specific extended functionality.*
- *Host tested by analysing traffic and looking for erroneous data from the host:*
  - *Host is important to test to reduce obvious errors and have a common error catch mechanism.*

Chapter 4 − WORK UNDERTAKEN

4.1 TEST METHODOLOGY

The test methodology describes the approaches taken to test IG and host solutions to ensure their compliance to the CIGI standard. Annex A describes the design process by which the original test methodology was formed by the Dstl led industry team that developed it.

There are many different ways of testing implementations of standards, ideally the test process for compliance would be fully automated, contain only objective tests and give clear explanations for any point of failure. Automated testing requires a clear definition of input vs. output, as CIGI has no such clear definition of exactly what the output is an automated approach is largely unachievable, requiring many of the test results to be manually determined. Objective testing in CIGI is also complicated as many of the potential use cases in CIGI are open to IG vendor interpretation (e.g., cloud top scudding). Because fully objectivity testing cannot be achieved the tests must therefore focus on the objective aspects of each use case (e.g., when changing the ratio of cloud top scudding from 0% – 50% – 100% the effect is seen to go from nothing-half-full).

For IG compliance testing a unit testing approach provides the best analogy for what is needed but given the need for manual verification the approach requires some modification. Each test would still focus on a single feature / use case, however a setup / tick / teardown approach was needed, with the tick only completing once validation had occurred.

Host testing is more complicated than IG testing and CIGI compliance means something different. A host is compliant with CIGI if all the packets that are sent comply with the specific rules for each parameter (e.g., range limits). Additionally checks can be made on expected packet size, validity of references etc. however some tests although logical (e.g., checking that an entity exists before attaching a view to it) are not actually stipulated by CIGI. This leads to a situation where some tests may be used to enforce compliance whereas others may exist to help explain why the IG may not be behaving as expected.

4.2 DERIVING USE CASES

During discussions an approach suggested by Boeing was to extract test cases directly from text following all of the ‘shall’ statements in the ICD, the implication being that anything the must be adhered is preceded by shall. While this would test the enforced parts of the specification it was thought that this assumes that the specification is completely correct with respect to the placement of shall statements. This would likely leave many obvious areas of CIGI compliance empty.

The UK MOD led STACI study (that preceded MSG-118) instead promoted an approach where use cases were extracted from the CIGI ICD from both the packet descriptions and the preceding sections where processes (e.g., system start-up and shutdown) and general considerations are described.

The packet descriptions in the CIGI ICD were used as the primary source of use cases (and the preceding sections as secondary) because the packet content infers all of the possible functionality that CIGI provides, the preceding sections only covering some of the available functionality.

The extraction of use cases from the packet definitions could also lead to holes in the standard that need to later be addressed, indeed during use case extraction a few ambiguous cases were uncovered which were reported to the SISO CIGI PSG for correction.

The process for use case extraction goes through each parameter of a packet identifying all possible uses for that parameter. These may be directly described in the packet text or further described in the preceding sections (e.g., as with the start-up / shutdown sequences for IG Control). By doing this for each parameter the superset of all use cases for all parameters will be obtained.

Annex C describes in detail the refined methodology that was originally designed and agreed for MSG-118.

A major difference between the UK MOD design and the MSG-118 approach was the removal of the domain / fidelity levels concept as this was considered to be too subjective and not necessary at the early stages of compliance (and could be later added as an overlay to the tests). Although the updated process was initially effective it was very time heavy and error prone. The process was further simplified to expedite the use case extraction and test case specification prior to implementation, the final process being:

- For each packet:
  - Determine use cases (column A).
- For each use case:
  - Determine required tests.
  - For each test:
    - Determine test steps (column E); and
    - Determine pass / fail criteria (column F).

This simplified approach removes test specific parameters (column C) and dependencies / related packets (column H) and also defers the identification of test setups (column D) and dependencies (column H). This removes a lot of complexity and overhead while providing a better separation between the test to be implemented and the implementation details of the test. The deferred columns could then be identified during implementation, drawing on the pool of test setups that had already been implemented for previous tests.

Also determined for each packet was a set of defaults that would be assumed for each packet, this prevents the need for assumptions later during implementation. An example of the most recent use case spreadsheet can be seen in Figure 4-1.

	A	B	C	D
1	Id	Use Case / Constraint Description	Test Setup Reference	Test Steps
2	IG Control (IGC)			
3	U1.1	Put IG into operate mode (startup)	TS2	
4	U1.2	Put IG into standby mode (shutdown)	TS2	1. Populate an IGC:IG = 0 (Reset/Standby). Because the IG Control and SOF packets are manditory in each data transmission there is no explicit send command needed.
5	U1.3	Enable global extrapolation	TS12	1. No further action.
6	U1.4	Disable global extrapolation	TS12	1. IGC:Smoothing Enable set to 0 (disable)

E	F	G
Pass / Fail Criteria	ICD Reference	Notes
1. IG Mode returned in SoF changes from 0 to 1 (Operate)	4.7	
1. IG SoF return IG Mode = 0 (Standby/Reset) **2.** IG is returned to default state (i.e. session state is reset)	4.7	
1. Entities / attached views (with extrapolating enabled & valid velocity / acceleration) move smoothly. For asynchronous operation a valid timestamp will need to be supplied by the Host for each data transmission.	6.1.1 Table 5	This has no effect unless extrapolation is supported.
1. Entities (with or without extrapolating enabled & valid velocity / acceleration) move only when entity position is updated (Entity Position) **2.** Attached views move only when attached entity position is updated (Entity Position) packet or view offset / attachment (View Control) is updated	6.1.1 Table 5	

F	G	H	I
ICD Reference	Notes	Packet Defaults	
4.7			
4.7		Major Version Database Number Entity Type Substitution Enable Minor Version IG Mode Timestamp Valid Smoothing Enable Host Frame Number Last IG Frame Number Timestamp	**4** **0** **0** **0** **0 (reset/standby)** **0** **0** **0** **0** **0**
6.1.1 Table 5	This has no effect unless extrapolation is supported.		
6.1.1 Table 5			

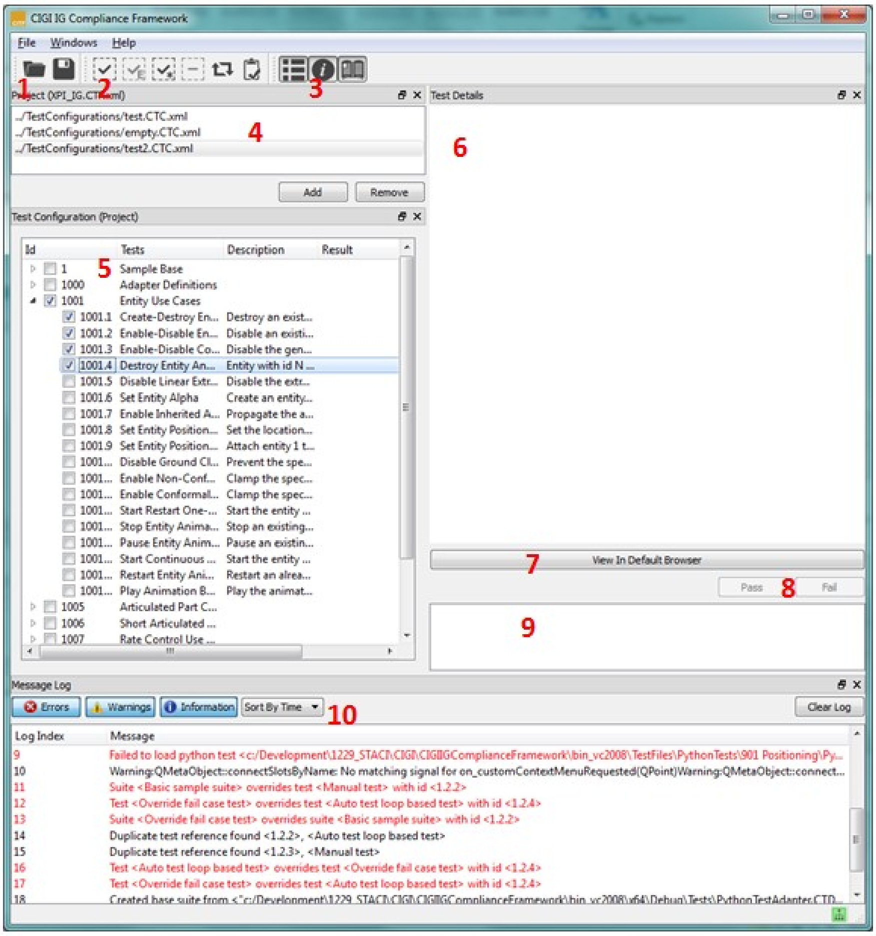
Figure 4-1: Most Recent Layout of Use Case Spreadsheet.

4.3 SOFTWARE DEVELOPMENT

As the tools had already been developed only small modifications were made based on experience from the team in using the tools.

4.3.1 IG Compliance Tool

The IG compliance tool is a GUI written in C++ using Qt and Boost that facilitates running the CIGI unit tests. Although it looks complex it is targeted at a developer end user. An example of the GUI can be seen in Figure 4-2.



1. Test configuration toolbar
2. Test toolbar
3. Window toolbar
4. Project Configuration
5. Test configuration hierarchy
6. Test details view
7. View test details in system default browser
8. Pass / fail test buttons
9. Test result comment box
10. Message log

Figure 4-2: IG Compliance Tool GUI.

When each test is run, the test details displays text and images that represent the expected visual output of the IG. The tool is hosted in the CIGI Compliance Tools SourceForge site [3], the current development being done under MSG-118 is hosted on a development branch [3]. The source repository contains the source for the projects, a prebuilt win32 version of the tools (in \CIGIIGComplianceFramework\bin\win32) and documentation for the compliance tool (in \CIGIIGComplianceFramework\docs).

The project contains both user documentation (for IG vendors running the tests) and developer documentation (for developers of new tests or maintainers of the software).

Although the IG compliance tool is written in C++ the tests are mostly expected to be written in Python using the PythonTestAdaptor. This is a special C++ test module that accumulates all of the python tests in a directory and makes them available to the GUI. Python is much more flexible than C++ in its distribution and also somewhat quicker to write. The only tests that are expected to be written in C++ are those that might require high performance or are very low level (like startup / shutdown).

4.3.1.1 Python Tests

Python tests are split into fixtures and tests. A python fixture is analogous to xUnit [5] fixtures in that it provides some kind of setup for the test. One notable difference however is that a fixture may require more than a single frame to complete (e.g., if loading a terrain), this then requires an additional setup function to be determined, that of the specific setup of the actual test (called deferredSetup in the python tests).

The general structure of tests is always the same so developers can quickly become accustomed to this minor complexity.

Because the python integration has been done using Boost Python it is not possible to run the python tests directly in a python debugger, because none of the CIGI python implementations exist except the C++ implementations that have been exposed programmatically. The tool could potentially be refactored at a future time to operate as a python only implementation (probably much simplified) but this was not considered necessary for the purposes of MSG-118.

The tool has been designed to be extended by adding extra python tests, C++ tests, data transports, tick handlers and report generators without requiring recompilation of the core application (i.e. via SDK's and configuration changes).

More details of the IG compliance tool design can be found in Annex A and also in the user and developer documentation in the SourceForge project directory.

4.3.2 Host Traffic Analyser

The host traffic analyser acts as a gateway between the host application and a single IG that intercepts the host->IG messages and performs a number of tests on them. A diagram of the data flow is shown in Figure 4-3.

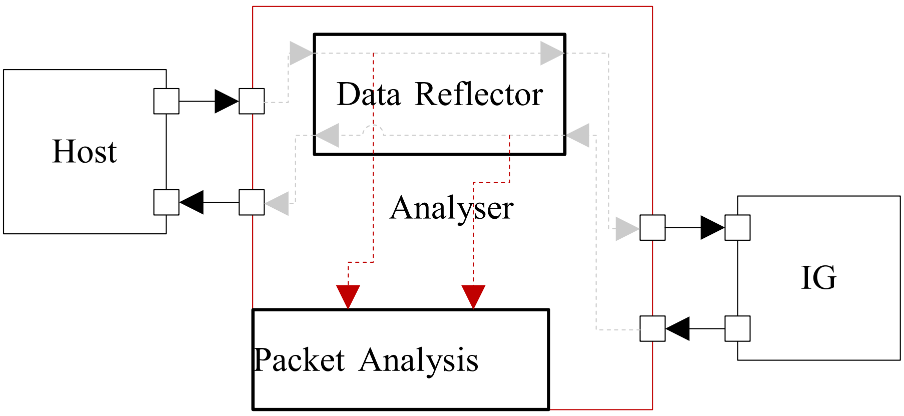


Figure 4-3: Host Traffic Analyser Conceptual Flow of Traffic.

Any errors captured during analysis are written to the GUI. A simple report is also generated once a session ends.

Figure 4-4 shows the host traffic analyser GUI with some example captured errors. Once an error is detected it can be viewed along with the associated invalid values, this allows the host application to be debugged to determine what the cause of the error might be. It is also possible to look at all of the packets for the current and previous messages to better determine the context of the error. In the displayed case (CIGI 3.3 host) entity 1001 has been sent with an out of range yaw value (must be 0 – 360). The same error has been received for the same entity 258 times.

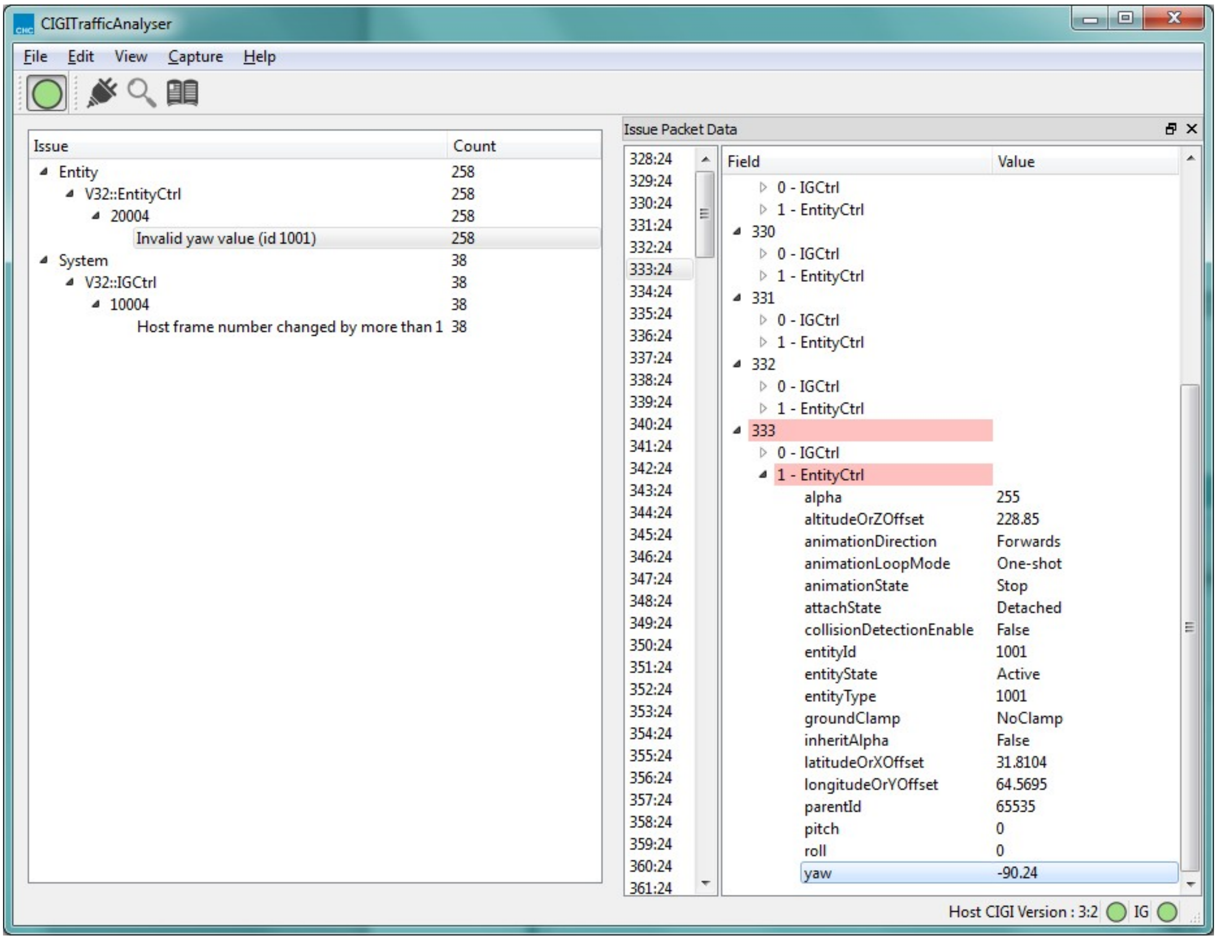


Figure 4-4: Host Traffic Analyser Example Showing Captured Errors.

The host traffic analyser can potentially support analysis of any CIGI version that conforms with the base packet definitions of CIGI 3.x or CIGI 4.x, it was originally developed against CIGI 3.3 but has been adapted to analyse CIGI 4.0 traffic directly. The analysis functionality is implemented by adding plugins to allow extension without needing to recompile to core application. It is also possible to implement additional data transport plugins, the default implementation is a simple UDP datagram data transport.

While some tests involve directly looking at packet data others require that a basic simulated IG state is maintained, e.g., the existence of entities / views in the simulation. This allows such things as the validity of references to be checked and is implemented in the analysis plugins where needed.

4.4 TEST DEVELOPMENT

Once the test cases have been specified from the use cases they can be implemented as python scripts. C++ can be used but the need for a compiler makes the development more complex and was not considered necessary.

First a suitable test setup needs to be found from the pool of already developed test setups. A test setup is a python script that sets up the starting conditions for a test, the test setups can be quite complicated involving many steps but crucially everything the test setup does should have been previously tested in smaller test cases, indeed some tests involve only calling the test setup with no specific test code at all (i.e. it tests only the test setup). If a test setup cannot be found from the pool then a new test setup has to be developed that may derive from an existing test setup, e.g., in the current version TS20 inherits from TS9 inherits from TS2 inherits from TS1 to generate a long and complex flow of commands prior to test execution.

4.4.1 Test Data

One of the issues discussed early in MSG-118 was how to generate or specify test data. Testing compliance with CIGI requires tests that need to hard-code some aspects of the moving models or terrains that it is based on. An attempt was made early on to try to describe the requirements of one of the simple moving models used in the initial UK MOD work and it was considered infeasible to attempt to do this for any more complex test model or terrain.

For terrains the additional complexity of coordinate systems and projection needed to be considered, the test terrain originally developed for the UK MOD work contained some useful features but was generated assuming a fixed UTM coordinate system that may be incompatible with many IGs. The model would therefore require reinterpreting which in itself could generate errors which would not be present in the IGs normal database generation process.

While these were considered valid concerns a decision was made to retain the existing test models and terrains in OpenFlight format as reference as OpenFlight is considered enough of an industry standard that most serious IG vendors will have some form of support for it. Where there is no support an effort would need to be made to generate an alternative that is close enough such that the tests do not require change.

4.4.2 Testing

Developing the python tests while initially complicated was relatively quick; the problem came when trying to ascertain how successfully the test had been implemented. As no CIGI V4.0 IG was available the only alternative was to use a CIGI 3.3 IG and develop a gateway that could convert the CIGI 4.0 traffic to CIGI 3.3 traffic. While this was quite simple to implement using the libCIGI library (see Ref. [6]) it would obviously only be able to provide a sufficient test for functionality that existed in CIGI 3.3, any features unique to CIGI 4.0 could not be tested and therefore any tests written would be unproven.

It was expected that soon after the release of CIGI 4.0 a number of IG vendors would adapt their implementations to support the standard but if work has been done little has been made of this in public. Without CIGI 4.0 IG's development and testing of the tests was very difficult, requiring a good level of python skill to implement and potentially being error prone due to the gateway.

The best approach that was finally put forward was to use the gateway in conjunction with the CIGI Multi-Purpose-Viewer (MPV) [7]. The MPV is not a production IG but a testbed developed by Boeing used for CIGI development. It supports a large number of 3.3 features and should do so in a way that would have been considered "compliant" with CIGI 3.3.

4.5 WHAT CIGI COMPLIANCE MEANS

Given the completion of the test implementations and an IG vendor generates a test profile for their IG, what does this give the audience requiring CIGI compliance? The end result is likely a pass result for a subset of a large number of tests that shows that the IG is conforming to that subset. The end user can then base their IG assessment on a number of criteria:

- The number of tests passed:
  - This is a very crude measurement and does not convey how an IG will meet a customer's specific needs; it only broadly conveys how many core CIGI features an IG has implemented.

- A comparison of the packets tested against those presumed needed:
    - If a project has identified a requirement for a packet that the IG does not support then this infers the IG will not meet the project needs, if the feature is provided but not in the CIGI standard way then it is non-compliant.
- Identifying precise subset of use cases / tests that the project needs and comparing:
    - For each project this may be complex and even infeasible as it is an indirect mapping of requirements to IG features which may be better done at a higher level initially; and
    - The domain / fidelity overlay approach discussed in Section 4.1 may aid in broad identification of tests required.

Selecting an IG that is fit for a single project or even a range of projects (as many companies will wish to do) is a very complex task. Without CIGI as there are many subjective elements that also need to be considered. CIGI compliant does not mean that it meets the visual requirements of a simulation project, for instance a CIGI compliant IG could be made that is entirely unrealistic. CIGI compliance only provides a level of confidence that CIGI commands will be interpreted correctly.

Chapter 5 − MEETINGS AND ORGANISATION

MSG-118 began with a 2 day face-to-face kick-off meeting starting on 18 April 2013. At this meeting the task background (including prior UK MOD funded STACI research) and objectives were presented and each country member gave a brief presentation. Some discussion was had on what contributions were planned from each country.

Following the kick-off meeting there was a 6-month period where very little happened in the MSG-118 Group. After a teleconference in October 2013 regular monthly teleconferences were planned to ensure that constant progress was being made in the group.

The use of WebEx as a videoconferencing platform was very effective and WebEx (or a similar alternative) is recommended for use in other collaborative tasks, most important was the ability to screen share and in some cases to pass control of the presentation / allowing other members to interact with the desktop.

The videoconferences served well as a discussion forum for developing the process for the group but as most participants only had a very small amount of time available outside of the teleconferences it was difficult to allocate development tasks to group members. Those who did have development time found that by the time the engineer had become accustomed to the work they had run out of budget.

A standardised template evolved from the monthly meetings that made it simpler for 1 person to both chair and minute the discussion. This was considered quite useful to capture the progress and actions to be completed however actions often went uncompleted for a long time.

One of the considered shortcomings of the group was probably the lack of involvement from key IG vendors. If some of these had been involved it might have encouraged more internal investment into V4.0 support as well as test beds for the compliance tools. As it was there was probably far too much representation from the part of the customer community and not enough from the part of the vendor.

Overall it is considered that the structure and frequency of the meetings would have been very effective had the members had considerably more time / budget available to dedicate to the technical development of the products. Without sufficient budget the meetings later in the task group mostly resulted in a failure to allocate any tasks to members.

Meeting minutes were put onto the MSG-118 SharePoint site and while WebEx recordings were saved it was not possible to upload these to SharePoint due to file size restrictions.

Chapter 6 – MSG-118 OUTPUTS

6.1 TEST TOOLS

The tools (as discussed in Chapter 4) are available on SourceForge. A "final" MSG-118 build needs to be created such that it can be easily downloaded and used by developers but given the interim state of the tests this may not be of significant benefit.

Additionally the libCIGI C++ library developed as part of the compliance tools is also publically available on SourceForge [2].

6.2 TESTS

In total 13 / 39 CIGI packets were considered complete in terms of use cases and test specifications with a further 14 still considered 'under review'. A total of 200 use cases have so far been identified.

Only a small number of tests have been implemented in python:

- IG Control: 8 tests;
- Entity control: 9 tests;
- HAT / HOT request: 6 tests;
- Line of sight segment: 16 tests;
- Line of sight vector: 14 tests;
- Articulated Part: 3 tests;
- Short Articulated Parts Control: 4 tests;
- View Control: 4 tests; and
- Position Request: 16 tests.

The python tests reside within the tools source control on SourceForge, specifically in the path \CIGIIGComplianceFramework\bin\TestFiles\. The folders of interest are FixturesV40 and PythonTestsV40.

6.3 DOCUMENTATION

A number of documents were generated by the MSG-118 Task Group, these included:

- Sample IG test definitions (Annex A of this document);
- Sample CIGI 3.3 compliance IG use cases (Annex B of this document);
- CIGI compliance tools test methodology (Annex C of this document);
- Test specifications spreadsheet (available from the MSG-118 SharePoint);
- Meeting minutes (available from the MSG-118 SharePoint);
- Meeting recordings (available from XPI Simulation); and
- Review report template (available from the MSG-118 SharePoint).

Chapter 7 – OUTSTANDING ISSUES / FURTHER WORK

7.1 OUTSTANDING ISSUES / ACTIONS

A number of outstanding issues remain from MSG-118:

1) Final build of MSG-118 version tools and tests.
2) Future development and maintenance of the toolset.
3) Clean up of test directories.
4) Open actions from MSG-118:
    - Each country to complete Form 13; and
    - Many others can be safely closed (as they are considered to have expired due to the close of the Group).

7.2 FURTHER WORK

Some items of work have been identified that would be beneficial to the compliance tools, namely:

1) Determine how compliance will be handled by the SISO CIGI PSG, i.e. how it will be registered.
2) Move SourceForge administration to SISO CIGI PSG.
3) Look again at Mozilla Public License 2 (MPL2) and possible move to Massachusetts Institute of Technology (MIT) / Berkeley Software Distribution (BSD) license.
4) Simplify test tool usage.

Chapter 8 – LESSONS LEARNED, CONCLUSIONS AND RECOMMENDATIONS

8.1 LESSONS LEARNED

During the 3 years that MSG-118 has been active a number of key lessons have been learned that should be considered by current and future Task Groups:

- Task Groups planning to develop a large technical product need to ensure that all parties can provide a technical contribution, including engineering effort.
- Use cases / test definitions should be clearly separated from implementation details.
- Use cases / test definitions should be kept concise, identifying key points and using notes to describe any special handling.
- Key stakeholders should be identified and engaged with early on.

8.2 CONCLUSIONS

- Tools are in a functional state if not completely user friendly.
- MSG-118 has not succeeded in completing the tests required to demonstrate compliance.
- Meeting frequency, structure and documentation were adequate but participants lacked the time / budget to work outside the meetings due to the volunteer nature of their participation.
- Compliance approach is sound but needs more effort to be usable.

8.3 RECOMMENDATIONS

The MSG-118 Task Group has had some mixed success and this is largely due to the large amount of technical content that needed to be generated and the limited budgets that group members had available to them to do this:

1) Future groups developing technical products need suitably qualified technical staff and sufficient budget, planning and awareness of cost prior to group formation as this is critical to success; ideally budget needs should be identified in the exploratory team process with countries committing to provide the agreed share of resources.
2) Product development and maintenance needs to be considered by SISO and NMSG, executable products have users and issues beyond the end of the task groups that create them:
    a) SISO and NMSG should have policies covering all kinds of product artefacts.
    b) SISO and NMSG should have managed public repository areas (e.g., SourceForge, GitHub) used to store and version technical content and product artefacts where these are evaluated as being suitable for public release to NATO industry and elsewhere.
    c) The use of standards within simulation and training have great benefit for interoperability and reuse, however the approach and cost of compliancy testing must be taken into account in business cases where use of standards are mandated.
3) Use virtual conferencing services that provide screen-share, transfer of control and video for regular Task Group meetings.

Chapter 9 − REFERENCES

[1] SISO-STD-013-2014 Common Image Generator Interface V4.0, https://www.sisostds.org/DigitalLibrary.aspx?Command=Core_Download&EntryId=42031, Accessed 26 June 2018.

[2] libCIGI C++ library SourceForge, https://sourceforge.net/projects/libcigi/, Accessed 26 June 2018.

[3] CIGI Compliance Tools SourceForge, https://sourceforge.net/p/cigicompliance/code/HEAD/tree/branches/2015_04_30%28NMSG%29_NewFixtureChanges/, Accessed 26 June 2018.

[4] MSG-118 Activity Page, https://www.sto.nato.int/search/Pages/activities_results.aspx?k=msg-118&s=Search%20Activities, Accessed 26 June 2018.

[5] xUnit Overview, https://www.martinfowler.com/bliki/Xunit.html, Accessed 26 June 2018.

[6] libCIGI Gateway Example, https://svn.code.sf.net/p/libcigi/code/trunk/Examples/SimpleGateway, Accessed 26 June 2018.

[7] CIGI Multi-Purpose Viewer, http://cigi.sourceforge.net/product_mpv.php, Accessed 26 June 2018.

Annex A – SAMPLE IG TEST DEFINITIONS

A.1 INTRODUCTION

Using the derived use cases (see Annex B), the following packets will have test definitions derived from them:

- IG Control (IGCtrl);
- Entity Control (EntityCtrl);
- Rate Control (RateCtrl);
- View Control (ViewCtrl);
- Celestial Sphere Control (CelestialSphereCtrl);
- Articulated Part Control (ArticulatedPartCtrl);
- Short Articulated Part Control (ShortArticulatedPartCtrl);
- Weather Control (WeatherCtrl); and
- Atmosphere Control (AtmosphereCtrl).

A.1.1 Fidelity Levels

Fidelity levels currently range from 1 to 3 with 1 representing low fidelity simulation while 3 represents (at the time of writing) a high fidelity simulation. To comply with basic requirements for a domain / fidelity pairing an IG must satisfy all test cases for fidelity levels equal and below the required level, e.g.:

- For Air-3 all Air-3, Air-2 and Air-1 tests must be passed.

Some test cases do not apply for a given domain; in these cases no number will be assigned.

Table A-1: Sample IG Test Definitions.

Test Id	Description	Pass Criteria	Test Dependencies
1	IG Control (IGCtrl)		
1.1	Put IG in Operating Mode	• The start-up sequence is followed as described in Section 2.8 of Ref. [2].	
1.2	Put IG in Standby Mode	• The shutdown sequence is followed as described in Section 2.8 of Ref. [2].	1.1
1001	Entity Control (EntityCtrl)		
1001.1	Create-Destroy Entity	• An entity is seen to be created and destroyed.	1.1
1001.2	Enable-Disable Entity	• An entity is created and seen to be enabled and disabled.	1001.1
1001.3	Enable-Disable Collision Detection	• The collision detection is toggled throughout the test. • An entity is included in collision tests when collision detection is enabled. • The same entity is not considered in collision tests when collision detection is disabled.	1001.1
1001.4	Destroy Entity and Reuse ID for a New Entity	• An entity is created, destroyed, and re-created as a different type.	1001.1
1001.5	Disable Linear Extrapolation	• The linear extrapolation is toggled throughout the test. • When it is enabled, the entity should move smoothly. • When it is disabled, the entity should not move at all (as periodic position updates are not provided).	1001.1
1001.6	Set Entity Alpha	• The alpha will toggle between half and full alpha. • When the alpha is half, the entity should be semi-transparent. • When the alpha is full, the entity should be totally opaque.	1001.1
1001.7	Enable Inherited Alpha	• The parent entity's alpha will toggle between half and full alpha while the child entity's alpha is at half alpha. • When the parent entity's alpha is at half alpha the child entity should be 25% visible. • When the parent entity's alpha is at full alpha the child entity should be semi-transparent.	1001.6

Test Id	Description	Pass Criteria	Test Dependencies
1001.8	Set Entity Position (Unattached)	• An entity is positioned in the database at an expected location.	1001.1
1001.9	Set Entity Position (Attached)	• A moving entity has a child entity that maintains relative position.	1001.1
1001.10	Disable Ground Clamp	• An entity will travel across the database without clamping to the ground.	1001.1
1001.11	Enable Non-Conformal Ground Clamp	• An entity will travel across the database clamping to the ground but not orientating roll or pitch.	1001.1
1001.12	Enable Conformal Ground Clamp	• An entity will travel across the database clamping to the ground while orientating roll and pitch to the angle of the ground.	1001.1
1001.13	Start Restart One-Shot Entity Animation	• An animation will periodically restart. There should be a short duration between the end of the animation and when it is restarted to be able to confirm the animation was "one-shot".	1001.1
1001.14	Stop Entity Animation	• An animation will be stopped and restarted periodically. The animation will look unnatural as it will be interrupted half way through.	1001.1
1001.15	Pause Entity Animation	• A looping animation will be paused and un-paused periodically.	1001.16
1001.16	Start Continuous Entity Animation	• A looping animation will continue until the test is ended.	1001.1
1001.17	Restart Entity Animation	• A looping animation will be restarted half way through.	1001.16
1001.18	Play Animation Backwards	• An animation will be played backwards.	1001.1
1005	Articulated Part Control (ArticulatedPartCtrl)		
1005.1	Set Articulation Position	• Each of the different articulation types (x/y/z/roll/pitch/yaw) should be tested in turn.	1.1
1005.2	Set Articulation Partial Position	• All of the different articulation types will be set but only one type will be enabled at a time – resulting in the same effect as 1005.1.	1.1
1005.3	Disable Articulation	• An articulation will be toggled between enabled and disabled (not visible).	1.1
1006	Short Articulated Part Control (ShortArticulatedPartCtrl)		
1006.1	Set Articulation 1 x DoF	• Each different type of articulation (x/y/z/roll/pitch/yaw) will be tested in turn.	1001.1
1006.2	Set Articulation 2 x DoF	• Two of each different type of articulation (x/y/z/roll/pitch/yaw) will be tested in turn.	1001.1

Test Id	Description	Pass Criteria	Test Dependencies
1006.3	Disable 2 X Articulations	• Two articulations will toggle between enabled and disabled.	1001.1
1007	Rate Control (RateCtrl)		
1007.1	Set Top Level Entity Velocity	• Each type of rate (x/y/z/roll/pitch/yaw) will be tested in turn.	1001.1
1007.2	Set Local (body Relative) Entity Velocity	• An entity should travel in a large circle due to its local rate being set to forward and a slight yaw.	1001.1
1007.3	Set Attached Entity Velocity	• Each type of rate (x/y/z/roll/pitch/yaw) on a child entity will be tested in turn.	1001.1
1007.4	Set Articulation Velocity	• Each type of rate (x/y/z/roll/pitch/yaw) on an articulation will be tested in turn.	1001.1
1008	Celestial Sphere Control (CelestialSphereCtrl)		
1008.1	Set Time Of Day	• Various times of day will be cycled through.	1.1
1008.2	Disable Sky Ephemera	• Different ephemera (sun/moon/stars) are toggled enabled/disabled.	1.1
1008.3	Change Star Intensity	• Different star intensities are cycled through.	1.1
1008.4	Disable Continuous Time of Day	• The sun/moon/stars should remain static.	1.1
1008.5	Enable Continuous Time of Day	• The sun/moon/stars should move in real time.	1.1
1009	Atmosphere Control (AtmosphereCtrl)		
1009.1	Change Global Wind	• A particle effect entity should show the wind direction.	1001.1
1009.2	Change Global Visibility	• A database that has distance markers will be visible and the 1000 m and 2000 m distances should be toggled in visibility based on fogging.	1.1
1011	Weather Control (WeatherCtl)		
1011.1	Enable Global Weather Layer	• Global weather should be visible.	1.1
1011.2	Disable Weather Layer	• Weather should be toggled between enabled and disabled. • (Consider merging this in with 1011.1).	1011.1
1011.3	Enable Regional Weather Layer	• A region of weather is visible and the view cycles through positions in front/inside/ behind the region.	1001.1

Test Id	Description	Pass Criteria	Test Dependencies
1015	View Control (ViewCtrl)		
1015.1	Attach a View to an Entity	• Different combinations of view offset types (x/y/z/roll/pitch/yaw) will be cycled through.	1001.1
1015.2	Attach a View Group to an Entity	• The same effect as 1015.1 but the view is part of a group, and it is the group that the operations are applied to.	1001.1

Annex B – CIGI 3.3 COMPLIANCE IG USE CASES

B.1 INTRODUCTION

Following the strategy laid out in the design document the following packets will have use cases derived from them:

- IG Control (IGCtrl);
- Entity Control (EntityCtrl);
- Rate Control (RateCtrl);
- View Control (ViewCtrl);
- Celestial Sphere Control (CelestialSphereCtrl);
- Articulated Part Control (ArticulatedPartCtrl);
- Short Articulated Part Control (ShortArticulatedPartCtrl);
- Weather Control (WeatherCtrl); and
- Atmosphere Control (AtmosphereCtrl).

The use cases in the following section have been filled out to varying levels of detail, the original intention being that the existing use cases would be completely described and more use cases would be added as required, however this document is incomplete (as are the tools). So many of the tables are incomplete or even completely empty, awaiting further detail.

B.2 USE CASES

B.2.1 IG Control Use Cases

B.2.1.1 Parameters

Parameter	Default Value	Numeric Value
Major version	-	3
Database number	-	-
IG mode	Operate	1
Timestamp valid	Valid	1
Extrapolation / interpolation enable	Enable	1
Minor version	-	3
Byte swap magic number	-	-
Host frame number	-	-
Timestamp	-	-
Last IG frame number	-	-

B.2.1.2 Put IG in Operating Mode

Change the IG state from reset / standby to operating such that it can be used to display images.

Required Parameters	Parameter	Required Value
	IG mode	Operate
Pre-Conditions	– The IG must be returning SoF with ***IG mode*** set to *reset / standby*	

Required Parameters	Parameter	Required Value
	IG mode	Operate
Post-Conditions	– A **SoF** packet is returned from the IG with ***IG Mode*** set to *operate* – The default database is loaded o This may be no database at all	

B.2.1.3 Put IG in Standby Mode

Change the IG state from operate to reset / standby.

Required Parameters	Parameter	Required Value
	IG mode	Standby
Pre-Conditions	– The IG must be in the operate state	
Post-Conditions	– A **SoF** packet is returned from the IG with ***IG Mode*** set to *reset / standby* – All entities, effects, views etc. are removed from the IG scenegraph	

B.2.1.4 Enable Global Extrapolation

Enable extrapolation of entity and articulation positions for those entities that have the feature enabled.

Required Parameters	Parameter	Required Value
	Extrapolation / interpolation enable	Enable (1)
Pre-Conditions	– The IG must be in the operate state – Extrapolation must be disabled – Entity A is created that has extrapolation disabled – Entity B is created that has extrapolation enabled – Both entities have a velocity set (RateCtrl) – Both entities are visible – Both entities remain at their last set position	
Post-Conditions	– Entity A remains at its last set position – Entity B position extrapolates smoothly along the velocity vector set by RateCtrl – Entity B orientation rotates smoothly at a fixed rate o Any sudden changes in rotation due to gimbal lock are expected	

B.2.1.5 Disable Global Extrapolation

Disable extrapolation of entity and articulation positions for all entities.

Required Parameters	Parameter	Required Value
	Extrapolation / interpolation enable	Disable (0)
Pre-Conditions	– The IG must be in the operate state – Extrapolation must be enabled – Entity A is created that has extrapolation disabled – Entity B is created that has extrapolation enabled – Both entities have a velocity set **(RateCtrl)** – Both entities are visible – Entity A remains at its last set position – Entity B position extrapolates smoothly along the velocity vector set by RateCtrl – Entity B orientation rotates smoothly at a fixed rate – Any sudden changes in rotation due to gimbal lock are expected	
Post-Conditions	– Both entities remain at their last set position	

B.2.1.6 Switch to a Valid Existing Terrain Database

Change the terrain database to one that exists and is known to be valid.

<table>
<tr><td rowspan="2">Required Parameters</td><td>Parameter</td><td>Required Value</td></tr>
<tr><td>Database number</td><td>N</td></tr>
<tr><td>Pre-Conditions</td><td colspan="2">– The IG must be in the operate state</td></tr>
<tr><td>Post-Conditions</td><td colspan="2">– SoF packets are returned from the IG with the database number set to –N for an >= 0 number of frames
– Only SoF is returned from the IG
– Once the IG has completed loading the terrain a SoF packet is returned with the database number set to N</td></tr>
</table>

B.2.1.7 Switch to an Invalid or Missing Terrain Database

Change the terrain database to one that cannot be loaded.

<table>
<tr><td rowspan="2">Required Parameters</td><td>Parameter</td><td>Required Value</td></tr>
<tr><td>Database number</td><td>N</td></tr>
<tr><td>Pre-Conditions</td><td colspan="2">– The IG must be in the operate state</td></tr>
<tr><td>Post-Conditions</td><td colspan="2">– SoF packets are returned from the IG with the database number set to –N for an >= 0 number of frames
– Only SoF is returned from the IG
– Once the IG has determined that the terrain cannot be loaded a SoF packet is returned with the database number set to -128</td></tr>
</table>

B.2.2 Entity Control Use Cases

B.2.2.1 Parameters

Parameter	Default Value	Numeric Value
Entity ID	-	-
Ground / ocean clamp	No clamp	0
Inherit alpha	Not inherited	0
Entity state	Active	1
Attach state	Detach	0
Animation direction	Forward	0
Animation loop mode	One-shot	0
Animation state	Stop	0
Linear interpolation / extrapolation enable	Enable	1
Collision detection enable	Disabled	0
Alpha	Fully opaque	255
Entity type	-	-
Parent ID	-	-
Yaw / pitch / roll / latitude / longitude / altitude	-	-

B.2.2.2 Create Entity

Create an entity of a specific type that is enabled on creation.

Required Parameters	Parameter	Required Value
	Entity Id	-
	Entity State	Active (1)
	Entity Type	-
Pre-Conditions	– There must not be a pre-existing entity with the same EntityId – The IG must have a visual representation associated with the EntityType	
Post-Conditions	– The IG visual representation is created / loaded in good time and is visible	

B.2.2.3 Destroy Entity

Destroy an existing entity.

Required Parameters	Parameter	Required Value
	Entity State	Destroyed (2)
Pre-Conditions	– There must be a pre-existing entity with the same EntityId	
Post-Conditions	– The IG representation is completely removed from the scene	

B.2.2.4 Enable Entity

Enable a previously created entity that has been disabled.

Required Parameters	Parameter	Required Value
	Entity State	Active (1)
Pre-Conditions	– The entity must already exist – The entity must be disabled – Entity is not included in any line-of-sight or collision calculations	
Post-Conditions	– The IG visual representation must be visible – Entity is included in line-of-sight and collision calculations	

B.2.2.5 Disable Entity

Disable a previously created entity that has been enabled.

Required Parameters	Parameter	Required Value
	Entity State	Inactive (0)
Pre-Conditions	– The entity must already exist – The entity must be enabled – Entity is included in line-of-sight and collision calculations	
Post-Conditions	– The IG visual representation remains loaded / created but must not be visible – Entity is not included in any line-of-sight or collision calculations	

B.2.2.6 Enable Collision Detection

Enable the generation of collision detection responses involving the specified entity on an entity where it was previously disabled.

Required Parameters	**Parameter**	**Required Value**
	Collision detection enable	Enabled (1)
Pre-Conditions	– Entity collision detection is disabled o Entity is not included in any line-of-sight or collision calculations	
Post-Conditions	– Collision detection responses are sent when the entity intersects with another entity or the terrain	

B.2.2.7 Disable Collision Detection

Disable the generation of collision detection responses involving the specified entity on an entity where it was previously enabled.

Required Parameters	**Parameter**	**Required Value**
	Collision detection enable	Disabled (0)
Pre-Conditions	– Entity collision detection is enabled o Entity is included in line-of-sight and collision calculations	
Post-Conditions	– Collision detection responses are not sent when the entity intersects with another entity or the terrain	

B.2.2.8 Enable Non-Conformal Ground / Ocean Clamp

Clamp the specified entity to the terrain / ocean only affecting the height of the entity (i.e. longitude / latitude / roll / pitch / yaw remain are unaffected).

Required Parameters	**Parameter**	**Required Value**
	Ground / Ocean Clamp	Non-conformal (1)
Pre-Conditions	– Entity is positioned floating above the terrain / sea surface	
Post-Conditions	– Entity altitude is set to the height of the terrain / sea surface – The altitude is used as an offset to the conformed height o Positive values should offset above the surface (e.g. tank), negative values offset below the surface (e.g. submarine) – Latitude, longitude, pitch, roll and yaw remain unchanged	

B.2.2.9 Enable Conformal Ground / Ocean Clamp

Clamp the specified entity to the terrain / ocean affecting the height, pitch and roll of the entity to match the terrain surface (i.e. longitude / latitude / yaw remain are unaffected).

Required Parameters	**Parameter**	**Required Value**
	Ground / Ocean Clamp	Conformal (2)
Pre-Conditions	– Entity is positioned floating above the terrain / sea surface – Entity pitch and roll do not reflect the terrain surface orientation	
Post-Conditions	– Entity altitude is set to the height of the terrain / sea surface – The altitude is used as an offset to the conformed height o Positive values should offset above the surface (e.g. tank), negative values offset below the surface (e.g. submarine) – Pitch and roll of the entity (approximately) match the surface orientation – Latitude, longitude and yaw remain unchanged	

B.2.2.10 Disable Ground / Ocean Clamp

Prevent the specified entity from clamping to the terrain / ocean (i.e. longitude / latitude / altitude / roll / pitch / yaw remain are unaffected).

Required Parameters	Parameter	Required Value
	Ground / Ocean Clamp	No clamp (0)
Pre-Conditions	– Entity is positioned floating above the terrain / sea surface – Entity pitch and roll do not reflect the terrain surface orientation	
Post-Conditions	– Latitude, longitude, altitude, pitch, roll and yaw remain unchanged	

B.2.2.11 Enable Linear Extrapolation

Enable the extrapolation of entity position using provided velocity and acceleration.

Required Parameters	Parameter	Required Value
	Linear interpolation / extrapolation enable	Enabled (1)
Pre-Conditions	– Entity has velocity set for location and orientation (RateCtrl) – IGCtrl has extrapolation enabled	
Post-Conditions	– Entity moves smoothly along initial trajectory with rotation at constant rate	

B.2.2.12 Disable Linear Extrapolation

Disable the extrapolation of entity position using provided velocity and acceleration.

Required Parameters	Parameter	Required Value
	Linear interpolation / extrapolation enable	Disabled (0)
Pre-Conditions	– Entity has velocity set for location and orientation (RateCtrl) – IGCtrl has extrapolation enabled	
Post-Conditions	– Entity remains at the last set position regardless of velocity	

B.2.2.13 Set Entity Position (Unattached)

Set the location and orientation of an entity in geodetic coordinates.

Required Parameters	Parameter	Required Value
	Yaw / pitch / roll / latitude / longitude / altitude	-
Pre-Conditions	– Terrain with known reference location is loaded	
Post-Conditions	– Entity is located at the correct position on the terrain	

B.2.2.14 Set Entity Position (Attached)

Attach entity 1 to entity 2 and set the location and orientation of entity 1 in entity coordinates (i.e. relative to entity 2).

Required Parameters	Parameter	Required Value
	Yaw / pitch / roll / xOffset / yOffset / zOffset	-
	Attach State	Attach (1)
	Parent ID	-
Pre-Conditions	– Parent entity has been previously created – Attached entity does not have any rate	
Post-Conditions	– Attached entity is attached correctly relative to the parent entities coordinate frame (X north, Y east, Z down) – When the parent entity moves the position of the attached entity relative to the parent remains static	

B.2.2.15 Set Entity Alpha

Change the transparency of an entity.

Required Parameters	**Parameter**	**Required Value**
	Alpha	127
	Inherited Alpha	Not inherited (0)
Pre-Conditions	– Entity is created and visible – Entity is fully opaque	
Post-Conditions	– Entity appears 50% opaque	

B.2.2.16 Enable Inherited Alpha

Propagate the alpha of parent entity 1 to attached entity 2.

Required Parameters	**Parameter**	**Required Value**
	Alpha	127
	Inherited Alpha	Inherited (1)
	Attach State	Attach (1)
	Parent ID	-
Pre-Conditions	– Attached entity is created and visible – Attached entity is 50% opaque – Attached entity is not inheriting alpha – Parent entity is created and visible – Parent entity is 50% opaque – Entity appears 50% opaque	
Post-Conditions	– Entity appears 25% opaque	

B.2.2.17 Start / Restart One-Shot Entity Animation

Start the entity animation as a one-shot animation that stops animating when reaching the end of the animation.

Required Parameters	**Parameter**	**Required Value**
	Animation loop mode	One-shot
	Animation state	Start
Pre-Conditions	– Entity is created and visible	
Post-Conditions	– Animation starts – After the animation duration the animation stops – After the animation stops an ‘Animation Stop Notification’ packet is sent from the IG o The stop notification references the correct entity id	

B.2.2.18 Stop Entity Animation

Stop an existing animating entity from playing its animation.

Required Parameters	**Parameter**	**Required Value**
	Animation state	Stop
Pre-Conditions	– Entity is created and visible – Animation is playing	
Post-Conditions	– Animation stops o It is acceptable for an ‘out’ animation to be played for a short duration	

B.2.2.19 Pause Entity Animation

Pause an existing animating entities animation at the last displayed frame.

Required Parameters	**Parameter**	**Required Value**
	Animation state	Stop
Pre-Conditions	– Entity is created and visible – Animation is playing	
Post-Conditions	– Animation pause at the last displayed frame	

B.2.2.20 Start Continuous Entity Animation

Start the entity animation as a continuous animation. The animation should repeat forever.

Required Parameters	**Parameter**	**Required Value**
	Animation loop mode	Continuous (1)
	Animation state	Start
Pre-Conditions	– Entity is created and visible	
Post-Conditions	– Animation starts – After the animation duration the animation restarts – No 'Animation Stop Notification' packet is sent from the IG	

B.2.2.21 Restart Entity Animation

Restart an already playing entity animation. The existing animation position will be reset to the beginning of the animation.

Required Parameters	**Parameter**	**Required Value**
	Animation state	Start
Pre-Conditions	– Entity is created and visible – Animation is running	
Post-Conditions	– Animation restarts	

B.2.2.22 Play Animation Backwards

Play the animation in reverse order.

Required Parameters	**Parameter**	**Required Value**
	Animation direction	Backward (1)
Pre-Conditions	– Entity is created and visible – Animation is not playing	
Post-Conditions	– Animation plays backwards – When the animation reaches the beginning an 'Animation Stop Notification' packet is sent from the IG o The stop notification references the correct entity id	

B.2.2.23 Restart Entity Animation

Entity with id N is destroyed on frame F. On frame F+1 entity N is created with a different type.

This use case involves applying 0 on frame F to destroy an existing entity (specific by entity id N) followed by 0 on frame F+1 referencing the same entity id. The entity type must change between the existing and new entity instantiations.

Required Parameters	Parameter	Required Value
	Entity State	-
Pre-Conditions	– Entity of type A is created and visible	
Post-Conditions	– Entity of type B is created and visible	

B.2.3 Conformal Clamped Entity Control Use Cases

B.2.3.1 Parameters

Parameter	Default Value	Numeric Value

B.2.4 Component Control Use Cases

B.2.4.1 Parameters

Parameter	Default Value	Numeric Value

B.2.5 Short Component Control Use Cases

B.2.5.1 Parameters

Parameter	Default Value	Numeric Value

B.2.6 Articulated Part Control Use Cases

B.2.6.1 Parameters

Parameter	Default Value	Numeric Value
Entity ID	-	-
Articulated part ID	-	-
Articulated part enable	True	1
X offset enable	True	1
Y offset enable	True	1
Z offset enable	True	1
Roll enable	True	1
Pitch enable	True	1
Yaw enable	True	1
X offset	-	0.0
Y offset	-	0.0
Z offset	-	0.0
Roll	-	0.0
Pitch	-	0.0
Yaw	-	0.0

B.2.6.2 Set Articulation Position

Set the position of an articulation relative to its submodel coordinate frame.

Required Parameters	Parameter	Required Value
	Entity ID	-
	Articulated part ID	-
	X offset	Non-zero
	Y offset	Non-zero
	Z offset	Non-zero
	Roll	Non-zero
	Pitch	Non-zero
	Yaw	Non-zero
Pre-Conditions	– Entity already exists – Visual model contains an articulation for the specified ***articulated part id*** – Articulation specified by ***articulated part id*** supports all 6 degrees of freedom	
Post-Conditions	– Position of articulation is correct relative to submodel coordinate frame – Correct articulation is positioned – Position returned in **PositionResponse** (parent entity coordinate system) matches expected position	

B.2.6.3 Set Articulation Partial Position

Set the position of an articulation relative to its submodel coordinate frame.

Required Parameters	Parameter	Required Value
	Entity ID	-
	Articulated part ID	-
	X offset enable	True
	Y offset enable	False
	Z offset enable	True
	Roll enable	False
	Pitch enable	True
	Yaw enable	False
	X offset	Non-zero
	Y offset	Non-zero
	Z offset	Non-zero
	Roll	Non-zero
	Pitch	Non-zero
	Yaw	Non-zero
Pre-Conditions	– Entity already exists – Visual model contains an articulation for the specified ***articulated part id*** – Articulation specified by ***articulated part id*** supports all 6 degrees of freedom	
Post-Conditions	– ***X offset***, ***Z offset*** and ***pitch*** are applied – ***Y offset***, ***roll*** and ***yaw*** are ignored	

B.2.6.4 Disable Articulation

Prevent an articulated part from being displayed or interacted with.

Required Parameters	**Parameter**	**Required Value**
	Entity ID	-
	Articulated part ID	-
	Articulated part enable	False
Pre-Conditions	– Entity already exists – Visual model contains an articulation for the specified ***articulated part id*** – Articulation specified by ***articulated part id*** supports all 6 degrees of freedom – Articulation is enabled and visible	
Post-Conditions	– Articulation is no longer visible on entity – The geometry for the articulation is not included in any collision queries	

B.2.7 Short Articulated Part Control Use Cases

B.2.7.1 Parameters

Parameter	Default Value	Numeric Value
Entity ID	-	-
Articulated part ID 1	-	-
Articulated part ID 2	-	-
DoF select 1	Not used	0
DoF select 2	Not used	0
Articulated part enable 1	False	1
Articulated part enable 2	False	1
DoF 1	-	0.0
DoF 2	-	0.0

B.2.7.2 Set Articulation 1 x DoF

Set the position of an articulation relative to its submodel coordinate frame.

Required Parameters	**Parameter**	**Required Value**
	Entity ID	-
	Articulated part ID 1	1
	DoF select 1	X offset (1)
	Articulated part enable 1	True (1)
	DoF 1	Non-zero
Pre-Conditions	– Entity already exists – Visual model contains an articulation for the specified ***articulated part id 1*** – Articulation specified by ***articulated part id 1*** supports all 6 degrees of freedom – All 6 DoF are set to non-zero values	
Post-Conditions	– Position of articulation 1 is correct relative to submodel coordinate frame – Correct articulation is positioned – The other 5 DoF's remain unaffected – Position returned in **PositionResponse** (parent entity coordinate system) matches expected position	

B.2.7.3 Set Articulation 2 x DoF

Set the position of an articulation relative to its submodel coordinate frame.

Required Parameters	**Parameter**	**Required Value**
	Entity ID	-
	Articulated part ID 1	1
	Articulated part ID 2	2
	DoF select 1	X offset (1)
	DoF select 2	pitch (5)
	Articulated part enable 1	True (1)
	Articulated part enable 2	True (1)
	DoF 1	Non-zero
	DoF 2	Non-zero
Pre-Conditions	– Entity already exists – Visual model contains an articulation for the specified ***articulated part id 1*** – Visual model contains an articulation for the specified ***articulated part id 2*** – Articulation specified by ***articulated part id 1*** supports all 6 degrees of freedom – Articulation specified by ***articulated part id 2*** supports all 6 degrees of freedom – All 6 DoF for both articulations are set to non-zero values	
Post-Conditions	– Position of articulation 1 is correct relative to submodel coordinate frame – Position of articulation 2 is correct relative to submodel coordinate frame – Correct articulation is positioned for both articulations – The other 5 DoF's remain unaffected for both articulations – Position returned in **PositionResponse** (parent entity coordinate system) matches expected position for each articulation	

B.2.7.4 Disable 2 x Articulations

Set the position of an articulation relative to its submodel coordinate frame.

Required Parameters	**Parameter**	**Required Value**
	Entity ID	-
	Articulated part ID 1	1
	Articulated part ID 2	2
	Articulated part enable 1	False (0)
	Articulated part enable 2	False (0)
Pre-Conditions	– Entity already exists – Visual model contains an articulation for the specified ***articulated part id 1*** – Visual model contains an articulation for the specified ***articulated part id 2*** – Both articulations are enabled and visible	
Post-Conditions	– Both articulations are no longer visible on entity – The geometry for the articulations are not included in any collision queries	

B.2.8 Rate Control Use Cases

B.2.8.1 Parameters

Parameter	Default Value	Numeric Value
Entity ID	-	-
Articulated part ID	-	-
Apply to articulated part	False	0
Coordinate system	World / Parent	0
X linear rate	0.0	0.0
Y linear rate	0.0	0.0
Z linear rate	0.0	0.0
Roll angular rate	0.0	0.0
Pitch angular rate	0.0	0.0
Yaw angular rate	0.0	0.0

B.2.8.2 Set Top Level Entity Velocity

Set the velocity of an entity.

Required Parameters	**Parameter**	**Required Value**
	X linear rate	Non-zero
	Y linear rate	Non-zero
	Z linear rate	Non-zero
	Roll angular rate	Non-zero
	Pitch angular rate	Non-zero
	Yaw angular rate	Non-zero
Pre-Conditions	– Entity already exists – Entity has extrapolation enabled – Global extrapolation is enabled (0) – IGCtrl timestamp is valid and very closely approximates real clock time	
Post-Conditions	– Entity moves along the given velocity vector smoothly at a constant speed o There is no jump in entity position when assigning the velocity (i.e. movement starts smoothly from the last set position) – Entity orientation changes at a constant rate (for each axis) o Any sudden changes in orientation due to gimbal lock are expected o There is no jump in entity orientation when assigning the velocity (i.e. rotation starts smoothly from the last set orientation)	

B.2.8.3 Set Local (Body Relative) Entity Velocity

Set the velocity of an entity relative to its local coordinate frame (i.e. body relative in DIS). The example given below should result in a circular movement of the entity in the XY plane (i.e. driving in a circle) with a circumference of ~ 108 m and a period of 3.6 s.

Required Parameters	**Parameter**	**Required Value**
	Coordinate System	Local (1)
	X linear rate	3
	Yaw angular rate	10
Pre-Conditions	– Entity already exists – Entity has extrapolation enabled – Global extrapolation is enabled (0) – IGCtrl timestamp is valid and very closely approximates real clock time	

<table>
<tr><td>Post-Conditions</td><td>– Entity moves in a circular motion
– Entity returns to start position after 3.6 seconds
– After 100 loops (360 seconds) the entity returns to the start position exactly
– The velocity relative to the entity is the same regardless of whether it is a top level or attached entity</td></tr>
</table>

B.2.8.4 Set Attached Entity Velocity

Set the velocity of an entity.

<table>
<tr><td rowspan="7">Required Parameters</td><td>Parameter</td><td>Required Value</td></tr>
<tr><td>X linear rate</td><td>Non-zero</td></tr>
<tr><td>Y linear rate</td><td>Non-zero</td></tr>
<tr><td>Z linear rate</td><td>Non-zero</td></tr>
<tr><td>Roll angular rate</td><td>Non-zero</td></tr>
<tr><td>Pitch angular rate</td><td>Non-zero</td></tr>
<tr><td>Yaw angular rate</td><td>Non-zero</td></tr>
<tr><td>Pre-Conditions</td><td colspan="2">– Attached entity already exists
– Parent entity exists
– Entity has extrapolation enabled
– Global extrapolation is enabled (0)
– IGCtrl timestamp is valid and very closely approximates real clock time</td></tr>
<tr><td>Post-Conditions</td><td colspan="2">– Attached entity moves along the given velocity vector smoothly at a constant speed relative to the parent entity
    o There is no jump in attached entity position when assigning the velocity (i.e. movement starts smoothly from the last set position)
    o Movement is relative to the parent entity coordinate frame
– Attached entity orientation changes at a constant rate (for each axis) relative to the parent entity
    o Any sudden changes in orientation due to gimbal lock are expected
    o There is no jump in attached entity orientation when assigning the velocity (i.e. rotation starts smoothly from the last set orientation)
    o Orientation is relative to the parent entity coordinate frame</td></tr>
</table>

B.2.8.5 Set Articulation Velocity

Set the velocity of an articulation of an entity.

<table>
<tr><td rowspan="9">Required Parameters</td><td>Parameter</td><td>Required Value</td></tr>
<tr><td>Articulated part ID</td><td>-</td></tr>
<tr><td>Apply to articulated part</td><td>True (1)</td></tr>
<tr><td>X linear rate</td><td>Non-zero</td></tr>
<tr><td>Y linear rate</td><td>Non-zero</td></tr>
<tr><td>Z linear rate</td><td>Non-zero</td></tr>
<tr><td>Roll angular rate</td><td>Non-zero</td></tr>
<tr><td>Pitch angular rate</td><td>Non-zero</td></tr>
<tr><td>Yaw angular rate</td><td>Non-zero</td></tr>
<tr><td>Pre-Conditions</td><td colspan="2">– Entity already exists
– Entity has extrapolation enabled
– Global extrapolation is enabled (0)
– IGCtrl timestamp is valid and very closely approximates real clock time
– Visual model contains an articulation for the specified articulated part id
– Articulation specified by articulated part id supports all 6 degrees of freedom</td></tr>
</table>

Post-Conditions	– Articulation moves along the given velocity vector smoothly at a constant speed o There is no jump in articulation position when assigning the velocity (i.e. movement starts smoothly from the last set position) o Movement is relative to the articulations sub model coordinate frame – Articulation orientation changes at a constant rate (for each axis) o Any sudden changes in orientation due to gimbal lock are expected o There is no jump in articulation orientation when assigning the velocity (i.e. rotation starts smoothly from the last set orientation) o Orientation is relative to the articulations submodel coordinate frame

B.2.9 Celestial Sphere Control Use Cases

B.2.9.1 Parameters

Parameter	Default Value	Numeric Value
Hour	-	0
Minute	-	0
Ephemeris model enable	Enable	1
Sun enable	Enable	1
Moon enable	Enable	1
Star field enable	Enable	1
Date / time valid	Invalid	0
Date	-	01012000
Star field intensity	-	100

B.2.9.2 Set Time of Day

Changing the simulated time of day.

Required Parameters	**Parameter**	**Required Value**
	Hour	-
	Minute	-
	Date / time valid	Valid (1)
	Date	-
Pre-Conditions		
Post-Conditions	– For a varying range of times and dates the display of the sky should be approximately correct given the database latitude and longitude o E.g. 12:00 UTC should result in daylight in the UK and night time in Australia	

B.2.9.3 Disable Sky Ephemera

Disabling sky ephemera such as the sun, moon and stars.

Required Parameters	**Parameter**	**Required Value**
	Sun enable	Enable
	Moon enable	Enable
	Star field enable	Enable

Pre-Conditions	– The sun must be enabled and visible for the disable sun test – The moon must be enabled and visible for the disable moon test – The stars must be enabled and visible for the disable stars test
Post-Conditions	– When the sun is disabled it is no longer displayed in the scene at any time of day – When the moon is disabled it is no longer displayed in the scene at any time of day – When the stars are disabled they are no longer displayed in the scene at any time of day

B.2.9.4 Change Star Intensity

Changing the intensity of the stars to simulate limited visibility due to haze.

Required Parameters	**Parameter**	**Required Value**
	Star field intensity	75, 50, 25, 0
Pre-Conditions	– The stars must be enabled and visible – Star intensity must be 100%	
Post-Conditions	– The intensity of the stars is reduced according to the input values o As the correction may not be non-linear the result may not be half the visible brightness but a value of 0 should still result in no stars	

B.2.9.5 Disable Continuous Time of Day

Prevent the time of day from changing over the course of the simulation (except where explicitly set by the CelestialSphereCtrl).

Required Parameters	**Parameter**	**Required Value**
	Ephemeris model enable	Disable
Pre-Conditions	– The time of day must be set to a sunset scene – The sun must be visible on the horizon – The speed of time should be artificially increased using the **IGCtrl** ***timestamp*** field and multiplying by 100 o With continuous time of day enabled the sun should be visibly moving	
Post-Conditions	– The position of the sun (and all other ephemera) should remain static	

B.2.10 Atmosphere Control Use Cases

B.2.10.1 Parameters

Parameter	Default Value	Numeric Value
Atmospheric model enable	False	0
Global humidity	-	-
Global air temperature	-	-
Global visibility range	-	-
Global horizontal wind speed	-	0.0
Global wind direction	-	0.0
Global barometric pressure	-	-

B.2.10.2 Change Global Wind

Modify the winds global direction and speed.

	Parameter	Required Value
Required Parameters	Global horizontal wind speed	-
	Global wind direction	-
Pre-Conditions	– A scene entity must exist that is affected by wind (e.g. particle effect) o Any simulated drag on the entity (i.e. where the entity does not move at exactly the wind speed) should be disabled	
Post-Conditions	– As the speed and direction of the global wind changes the affected scene entity travels at the correct speed and direction o Direction = 0 => north o Direction = 90 => east o Direction = 180 => south o Direction = 270 => west	

B.2.10.3 Change Global Visibility

Modify the global visibility to simulate the effects of loss of visibility due to light scattering.

	Parameter	Required Value
Required Parameters	Global visibility range	-
Pre-Conditions	– A terrain must be loaded that contains explicit distance markers	
Post-Conditions	– As the visibility range is modified the correct visibility markers appear o At 1000 m visibility the 900 m marker should be visible, not the 1000 m o At 2000 m visibility the 1800 m marker should be visible, not the 2000 m	

B.2.10.4 Change Atmospheric Information

Modify the atmospheric information for generation of a physics based atmosphere model (and sensor support).

	Parameter	Required Value
Required Parameters		
Pre-Conditions		
Post-Conditions		

B.2.10.5 Enable Global Atmosphere Control

Modify the global visibility to simulate the effects of loss of visibility due to light scattering.

	Parameter	Required Value
Required Parameters		
Pre-Conditions		
Post-Conditions		

B.2.11 Environmental Region Control Use Cases

B.2.11.1 Parameters

Parameter	Default Value	Numeric Value

B.2.12 Weather Control Use Cases

B.2.12.1 Parameters

Parameter	Default Value	Numeric Value
Entity / region id	-	-
Layer id	-	-
Humidity	-	-
Weather enable	True	1
Scud enable	False	0
Random winds enable	False	0
Random lighting enable	False	0
Cloud type	None	0
Scope	Global	0
Severity	-	0
Air temperature	-	-
Visibility range	-	-
Scud frequency	-	0
Coverage	-	0
Base elevation	-	-
Thickness	-	0
Transition band	-	0
Horizontal wind speed	-	0
Vertical wind speed	-	0
Wind direction	-	0
Barometric pressure	-	-
Aerosol concentration	-	-

B.2.12.2 Enable Global Weather Layer

Apply a weather layer globally such that it exists in all locations at the specified elevation.

<table>
<tr><td rowspan="2">Required Parameters</td><td>Parameter</td><td>Required Value</td></tr>
<tr><td>Weather enable</td><td>True (1)</td></tr>
<tr><td>Pre-Conditions</td><td colspan="2">– An entity model must be loaded that represents distance markers tangential to the ellipsoid reference plane (i.e. parallel to the ground)
o Elevation must match the weather layer base elevation</td></tr>
<tr><td>Post-Conditions</td><td colspan="2">– The weather layer is visible
– The base of the weather layer is concurrent with the base of the reference entity</td></tr>
</table>

B.2.12.3 Disable Weather Layer

Disable a weather layer such that it is no longer displayed or considered in any other calculations.

Required Parameters	**Parameter**	**Required Value**
	Weather enable	False (0)
Pre-Conditions	– The weather layer must have been previously create and visible	
Post-Conditions	– The weather layer is no longer visible – Any calculations that use the localised sensor information (e.g. barometric pressure) should have compensated for the change	

B.2.12.4 Enable Regional Weather Layer

Apply a weather layer only within the specified environmental region.

Required Parameters	**Parameter**	**Required Value**
	Entity / region id	-
	Scope	Regional (1)
Pre-Conditions	– An entity model must be loaded that represents distance markers tangential to the ellipsoid reference plane (i.e. parallel to the ground) o Origin and orientation of the entity must match the region	
Post-Conditions	– Weather layer is confined to the regions volume	

B.2.12.5 Attach Weather Layer to an Entity

Attach a weather layer to the specified entity to simulate a specific weather phenomenon, e.g. cumulonimbus.

Required Parameters	**Parameter**	**Required Value**
	Entity / region id	-
	Scope	Entity (2)
Pre-Conditions	– An entity model must be loaded that represents distance markers tangential to the ellipsoid reference plane (i.e. parallel to the ground) o Origin and orientation of the reference entity must match the parent entity	
Post-Conditions	– Origin of weather volume is at the entity position – When the entity moves the weather volume moves with it	

B.2.12.6 Modify Scud Effect on a Weather Layer

Change the way that scudding is applied when passing through the weather (cloud) layer.

Required Parameters	**Parameter**	**Required Value**
	Scud enable	True (1)
	Scud frequency	-
Pre-Conditions	– Viewpoint is static within the cloud layer	
Post-Conditions	– As the view scud frequency changes the apparent density of the transition bands changes o 0 => no scud effect in transition band o 100 => completely solid in transition band o 50 => 50% coverage in transition band	

B.2.12.7 Modify Wind Within a Weather Layer

Change the way that scudding is applied when passing through the weather (cloud) layer.

Required Parameters	**Parameter**	**Required Value**
	Horizontal wind speed	-
	Vertical wind speed	-
	Wind direction	-
Pre-Conditions	– An entity (b) model must be loaded that represents distance markers tangential to the ellipsoid reference plane (i.e. parallel to the ground) o Origin of the entity must match the weather layer – An entity (b) model must be loaded that represents distance markers in elevation o Origin of the entity must match the region – A scene entity must exist that is affected by wind (e.g. particle effect) o Any simulated drag on the entity (i.e. where the entity does not move at exactly the wind speed) should be disabled	
Post-Conditions	– As the speed and direction of the global wind changes the affected scene entity travels at the correct speed and direction o Direction = 0 => north o Direction = 90 => east o Direction = 180 => south o Direction = 270 => west – As the vertical wind speed changes the affected scene entity travels at the correct speed and direction o E.g. a particle emitter with a constant emit velocity of 10 m/s in elevation should stay still when a vertical wind speed of -10 m/s is applied – The wind outside the volume of the weather layer is unaffected by the weather layer wind settings	

B.2.12.8 Modify Coverage of a Weather Layer

Change the coverage of the weather layer.

Required Parameters	**Parameter**	**Required Value**
	Coverage	-
Pre-Conditions	– Viewpoint is on the ground looking at the cloud layer	
Post-Conditions	– Coverage 0 => no visible weather layer – Coverage 50 => 50% visibility through weather layer – Coverage 100 => no visibility through cloud layer	

B.2.12.9 Modify Weather Layer Volume

Change the volume definition of the weather layer.

Required Parameters	**Parameter**	**Required Value**
	Thickness	-
	Transition band	-
Pre-Conditions	– Scudding is enabled and scudding frequency set to 100	
Post-Conditions	– Thickness increases the vertical height of the weather layer from the base height (i.e. top = base + thickness) – Transition band adds a graduated region outside the thickness volume	

B.2.12.10 Modify Visibility Within a Weather Layer

Change the visibility when looking through the weather layer.

Required Parameters	**Parameter**	**Required Value**
	Visibility range	-

Pre-Conditions	– Weather layer is completely solid
	– An entity model must be loaded that represents distance markers tangential to the ellipsoid reference plane (i.e. parallel to the ground)
	    o Origin of the entity must match the weather layer
	    o Multiple instantiations should be created at different heights
	    o All instantiations should be visible from the viewpoint (with no weather layer)
	– No transition band
	– Thickness 100 m
	– Viewpoint in centre of weather layer (i.e. 50 m above base)	

Post-Conditions	– When changing the visibility range the markers visibility should be changed accordingly
	    o With visibility at 100 m
	        ▪ The markers within the cloud range: should display the 80 m marker but not 100 m
	        ▪ Markers 100 m under the base elevation: should be able to see both 80 m and 100 m markers	

B.2.12.11 Modify Base Elevation of a Weather Layer

Display random lighting effects on the cloud layer.

Required Parameters	**Parameter**	**Required Value**
	Base elevation	-
Pre-Conditions	– An entity model must be loaded that represents distance markers in elevation – No transition band – Coverage 100	
Post-Conditions	– The base elevation of the weather layer matches the marker on the reference model	

B.2.12.12 Enable Random Lightning Effects Within a Weather Layer

Display random lighting effects on the cloud layer.

Required Parameters	**Parameter**	**Required Value**
	Random lighting enable	True (1)
	Severity	-
Pre-Conditions	– Weather layer id is a cloud layer – Cloud type is valid and of lighting producing type	
Post-Conditions	– Severity 0 => no lightning – Severity 5 => frequent lighting	

B.2.12.13 Set the Cloud Type of a Cloud Weather Layer

Set the type of cloud that the weather layer represents.

Required Parameters	Parameter	Required Value
	Cloud type	-
Pre-Conditions	– Layer id is ‘cloud layer 1’, ‘cloud layer 2’ or ‘cloud layer 3’	
Post-Conditions	– As the cloud type cycles through the standard types (1 - 10) the visual representation of the cloud approximates the appearance of the layer	

B.2.12.14 Modify Atmospheric Data Within a Weather Layer

B.2.12.15 Enable Random Wind Variation Within a Weather Layer

B.2.13 Maritime Surface Conditions Control Use Cases

B.2.13.1 Parameters

Parameter	Default Value	Numeric Value

B.2.14 Wave Control Use Cases

B.2.14.1 Parameters

Parameter	Default Value	Numeric Value

B.2.15 Terrestrial Surface Conditions Control Use Cases

B.2.15.1 Parameters

Parameter	Default Value	Numeric Value

B.2.16 View Control Use Cases

B.2.16.1 Parameters

Parameter	Default Value	Numeric Value
View ID	-	0
Group ID	None	0
X offset enable	Enable	1
Y offset enable	Enable	1
Z offset enable	Enable	1
Roll enable	Enable	1
Pitch enable	Enable	1

Parameter	Default Value	Numeric Value
Yaw enable	Enable	1
Entity ID	Ownship	0
X offset	-	0.0
Y offset	-	0.0
Z offset	-	0.0
Roll	-	0.0
Pitch	-	0.0
Yaw	-	0.0

B.2.16.2 Attach a View to an Entity

Attach a specific view to an entity in the scene.

Required Parameters	**Parameter**	**Required Value**
	View ID	0
	Entity ID	1
	X offset	Non-zero
	Y offset	Non-zero
	Z offset	Non-zero
	Roll	Non-zero
	Pitch	Non-zero
	Yaw	Non-zero
Pre-Conditions	– The entity must have been previously created and enabled – The view must be bound to a window output on the IG	
Post-Conditions	– The entity should be visible in the view – As the entity moves it should remain static in the view (i.e. world appears to move rather than entity, like being the driver of a car)	

B.2.16.3 Attach a View Group to an Entity

Attach a view group to an entity in the scene.

Required Parameters	**Parameter**	**Required Value**
	Group ID	1
	Entity ID	1
	X offset	Non-zero
	Y offset	Non-zero
	Z offset	Non-zero
	Roll	Non-zero
	Pitch	Non-zero
	Yaw	Non-zero
Pre-Conditions	– The entity must have been previously created and enabled – A view (0) must have been bound to the view group in **ViewDefinition** – The view (0) must be bound to a window output on the IG – The view must be offset from the group as in 0	
Post-Conditions	– The entity should be visible in the view – As the entity moves it should remain static in the view (i.e. world appears to move rather than entity, like being the driver of a car)	

B.2.17 Sensor Control Use Cases

B.2.17.1 Parameters

Parameter	Default Value	Numeric Value

B.2.18 Motion Tracker Control Use Cases

B.2.18.1 Parameters

Parameter	Default Value	Numeric Value

B.2.19 Earth Reference Model Definition Use Cases

B.2.19.1 Parameters

Parameter	Default Value	Numeric Value

B.2.20 Trajectory Definition Use Cases

B.2.20.1 Parameters

Parameter	Default Value	Numeric Value

B.2.21 View Definition Use Cases

B.2.21.1 Parameters

Parameter	Default Value	Numeric Value

B.2.22 Collision Detection Segment Definition Use Cases

B.2.22.1 Parameters

Parameter	Default Value	Numeric Value

B.2.23 Collision Detection Volume Definition Use Cases

B.2.23.1 Parameters

Parameter	Default Value	Numeric Value

B.2.24 HAT / HOT Request Use Cases

B.2.24.1 Parameters

Parameter	Default Value	Numeric Value

B.2.25 Line of Sight Segment Request Use Cases

B.2.25.1 Parameters

Parameter	Default Value	Numeric Value

B.2.26 Line of Sight Vector Use Cases

B.2.26.1 Parameters

Parameter	Default Value	Numeric Value

B.2.27 Position Request Use Cases

B.2.27.1 Parameters

Parameter	Default Value	Numeric Value

B.2.28 Environmental Conditions Request Use Cases

B.2.28.1 Parameters

Parameter	Default Value	Numeric Value

B.2.29 Symbol Surface Definition Use Cases

B.2.29.1 Parameters

Parameter	Default Value	Numeric Value

B.2.30 Symbol Text Definition Use Cases

B.2.30.1 Parameters

Parameter	Default Value	Numeric Value

B.2.31 Symbol Circle Definition Use Cases

B.2.31.1 Parameters

Parameter	Default Value	Numeric Value

B.2.32 Symbol Line Definition Use Cases

B.2.32.1 Parameters

Parameter	Default Value	Numeric Value

B.2.33 Symbol Clone Use Cases

B.2.33.1 Parameters

Parameter	Default Value	Numeric Value

B.2.34 Symbol Control Use Cases

B.2.34.1 Parameters

Parameter	Default Value	Numeric Value

B.2.35 Short Symbol Control Use Cases

B.2.35.1 Parameters

Parameter	Default Value	Numeric Value

Annex C – CIGI COMPLIANCE TOOLS TEST METHODOLOGY

THIS ANNEX IS A REPRODUCTION OF AN ORIGINAL DOCUMENT THAT WAS AN INFORMAL ARTIFACT OF THE DSTL STACI PROJECT (NOT A RELEASED DOCUMENT), HENCE IT IS INCLUSION AS AN ANNEX AND NOT A REFERENCED DOCUMENT.

Table of Contents

Glossary of Terms, Acronyms and Abbreviations

CCL	CIGI Class Library
CIGI	Common Image Generator Interface
COTS	Common Off The Shelf
GUI	Graphical User Interface
ICD	Interface Control Document
IEEE	Institute of Electrical and Electronics Engineers
IG	Image Generator
libCIGI	C++ CIGI interface library that supports V4.0
MSG	Modelling and Simulation Group
NATO	North Atlantic Treaty Organisation
SISO	Simulation Interoperability Standards Organisation
UDP	User Datagram Protocol

References and Publications

[1] L. Durham, W.Phelps: "CIGI Interface Control Document 3.3", 2008.
http://sourceforge.net/projects/cigi/files/CIGI%20ICD/Version%203.3/CIGI_ICD_3_3.pdf/download
[2] CIGI Class Library
http://cigi.sourceforge.net/product_ccl.php
[3] xUnit Wikipedia Page
http://en.wikipedia.org/wiki/XUnit
[4] libCIGI
https://sourceforge.net/p/libcigi/wiki/Home/
[5] CIGI Compliance Tools
http://sourceforge.net/p/cigicompliance/wiki/Home/
[6] Wireshark Network Analysis Tool
https://www.wireshark.org/

Acknowledgements

This document is an artefact of the North Atlantic Treaty Organisation (NATO) Modelling and Simulation Group (MSG)-118 activity aimed at providing compliance testing capabilities for the Common Image Generator Interface (CIGI) V4.0 SISO (Simulation Interoperability Standards Organisation) standard.
This work was originally based on efforts made by the UK MoD and Dstl to both promote and support the use of CIGI by providing the original CIGI compliance tools.

INTRODUCTION

General

Although CIGI has been a proprietary open format for over 10 years there has been no (publicly available) effort to test for compliance to the interface control document (ICD) during that period. As the original aim of CIGI was to ensure that the developers of systems incorporating IG's was easier and more uniform this was not a significant issue as little re-learning was required when switching between IG vendors.

Although the term 'CIGI compliant' has been used previously it is generally understood that this means very little because there is nothing to prove otherwise.

In existing CIGI implementations it is assumed / known that there are many deviations from the ICD from both the IG and the implementation of the CIGI host (whether accidental or forced due to IG non-compliance).

Customer Need

The Terms of Reference for this deliverable state that 'XPI shall design a prototype CIGI compliancy test framework according to the latest CIGI draft standard. The framework design should be flexible and extensible to take account of the variable nature of user defined and component packets as well as non-variable packets within the standard. The framework should be extensible and must have sufficient detail for a subset of the CIGI standard including the most used packet types to be used for test and experimentation within this project. XPI shall produce a report describing the compliancy framework design'.

Need for Compliance Testing

Simulation systems have acceptance tests run against them that cover as many of the expected use cases as possible. While these tests should cover the simulator working as expected it does not necessarily infer that the standards upon which the system is built are being adhered to. E.g.

> Many systems publish DIS Entity State PDUs. The Entity State PDU contains a large number of parameters and extensible fields. To be compliant the publisher should ensure that all parameters contain correct values.
>
> Many subscribers to the Entity State PDU will only consider a small number of the parameters, the rest are ignored as they are not required for the acceptance of the system. Therefore without compliance there is no way of ensuring that a publisher generates a valid PDU.

It may be considered that as long as a system passes the acceptance test that it is OK to be non-compliant. This may well be the case but it is still important to understand and record ***how*** the system is non-compliant so that future upgrade issues can be predicted and a plan made for addressing the shortcomings.

Now that CIGI is to become an official SISO standard there is a requirement to prove compliance against the standard as the perceived scope of CIGI is no longer just to make the system developers initial implementation simpler but also to ensure that maintenance and upgrade of systems is possible.

Scope of Compliance Testing

Testing of both host and IG implementations of the CIGI ICD are to be considered. The current scope of testing has been limited by the time available in the current package of work but a full implementation of the test suite should include most if not all of the CIGI interface.

For host compliance the scope is limited to the direct content within the CIGI packets sent by the host to the IG. No other application (e.g. a tank visual model selected rather than the visual model for a missile, which is then fired from an aircraft) will be considered as this is a feature of the simulation system design / implementation.

For IG systems both the content of the CIGI packets sent from the IG to the host and the visual output of the IG will be considered. Where terrains / moving models are required for a specific test these are intended to be supplied with the test suite to ensure that there is little ambiguity in the visual result. Any models provided will need to be converted to the internal IG format.

The method of data transport is assumed to be UDP unicast with an entire frame of data being sent as a single UDP packet.

The test configuration is a single host machine connected to a single IG on a private / peer-to-peer network.

TOOL REQUIREMENTS

The most significant features / requirements of any compliance tool are to ensure that

- the compliance of the target system to the standard is tested
- tests cover as much of the standard as possible
- any non-compliant features are reported to the user
- a standards committee is given responsibility for certifying IG solutions

Although these are the minimum requirements there are also some other desirable features of a compliance test suite.

- Testing can be performed quickly
- Testing / verification is automated
- Tests are objective
- Tools are freely available
- Tools are extensible
- Full source / documentation is available
- Tools are maintained in step with updates to the CIGI standard
- Testing can be performed directly by a customer prior to acceptance of a system
- Tools are capable of running on multiple platforms

With these features in place a compliance testing suite can be used during the development of a product rather than just at the point of testing. This ensures that developers see a benefit in the using the test suite earlier and that any issues are detected long before any official testing.

TEST STRATEGIES

There are many tests strategies that have been developed for varying systems and standards. Techniques such as manual inspection can be valuable where no other strategy is applicable. There are two strategies that have been considered for CIGI compliance, direct frame data analysis and unit testing.

Direct Frame Data Analysis

The contents of the data sent from the host to the IG and the IG to the host are interpreted by a 3rd party application and the contents tested for validity. This has the advantage that it can be run on an IG system during normal operation and so cover all the potential use cases of the IG & host.
While captured frame data can be manually inspected it is considered that this is of little benefit and the majority of benefit will come from applying automatic validation tests based on the constraints laid out in the CIGI ICD [1].
It is considered that direct frame analysis will provide all of the compliance testing for CIGI hosts. Many automatic checks can be derived from the ICD.

Unit Testing

Unit testing involves the development of a number of independent test cases that can be run in isolation to test the correct implementation of a single features or class of features. As unit tests require a controlled testing environment they are unsuited to CIGI host compliance but very well suited to IG compliance testing.
The benefit of unit testing for an IG is that tests can focus on a single feature relatively easily and when the feature is non-compliant the report can clearly show this.
The major disadvantage of IG testing is that the majority of output is visual and requires manual visual verification. Although when the output is completely defined it is possible to record reference images and perform an automatic image comparison this is not possible with CIGI as there is enough allowable variation between IG's that this is not possible (e.g. lighting, fogging, reflections).
When selecting pass / fail criteria the most quickly and accurately interpreted form should be used, in many cases a written description will suffice but for others a picture or even video should be used as reference.

HOST COMPLIANCE TESTING

The host compliance tool will rely on direct frame data analysis. To be an effective tool a number of automatic tests will need to be applied.

Tests

After reviewing the content of the CIGI packets the following test types have been identified

- Expected size
- Valid parameter ranges
- Reference validity (e.g. referenced entity exists)
- Expected extension (from a list of expected id, size definitions)
- Expected component (e.g. component validity for entity type as inferred by entity id)

It is considered that tests involving extensions and components can only be tested with modification to configuration files of the test suite, as these definitions of the extensions will change per project and per implementation.

Custom Tests

Support for custom tests (i.e. those written by IG or host vendors) would provide benefit for both internal development and customer compliance tests. Care must be taken however that the standard packet tests are not in any way affected by the extensions.
It may not be possible in the planned implementation of the test suite to guarantee this. A simple solution to this is to chain the standard and extended compliance tools together so that both are run simultaneously. Using this method the extended version can be validated against the standard version.

Expected Size

This involves checking the reported packet size against that specified in the ICD. E.g. for *'Entity Control'*
id 2
size 48
When dealing with variable length packets (containing records) such as *'Symbol Line Definition'* the size needs to be compared against the formula

$$S = a + bc$$

Where
S = final size
a = base size (bytes)
b = record size (bytes)
c = number of records
In the case of the '*Symbol Line Definition* 'packet this formula would become

$$S = 16 + 8n$$

Where
n = number of vertices in the symbol.

Valid Parameter Range

Many of the parameters in CIGI packets define a specific range within which they are valid. The CCL [2] implements automatic bounds checks that validate many of these but requires IG / host code to be written to support handling violations.
When a packet is received the valid range for each parameter that defines one will be checked, e.g. *'Entity Control:yaw'* must be between +0.0 and +360.0.

Note that the above example does not cover the erroneous case where the angle is supplied in radians rather than degrees (as the range would be +0.0 to ~6.2832). This case should however be covered by standard system acceptance tests.

Reference Validity

Packets such as *'Entity Control'* reference other scene objects in their parameters. In the example of *'Entity Control'* (when *AttachState* = 1) the entity referenced by *ParentId* must refer to a valid entity on the IG.

Expected Extension

In the simple case this test can compare a non-standard packet id with a list of known valid extensions. This requires only a simple extension to the configuration to add the list of valid ids.

Extended Checking

More complex tests would involve support for custom packet handling so extension data could be analysed and have tests 0 through 0 applied.

Expected Component

Expected component tests have a simple case of comparing against a list of valid ids (similar to extension checking). However because components are further subdivided into classes a list would need to be provided for each case.

Extended Checking

To implement more complex analysis would require support for custom component packet handling.

Redundant Parameters

In some cases some parameters within a packet are considered redundant (e.g. when EntityControl:EntityState is set to Destroyed(2)). The tests need to support this and only report errors on data that is not redundant.
When the IG is set to the Standby/Reset mode all packets parameters except for IG Control should be considered redundant. However if packets (other than IG Control) are received while the IG is in Standby/Reset an error should be reported.

Intercepting Frame Data

There are two approaches considered for intercepting frame data sent from the CIGI host.

COTS Data Logger

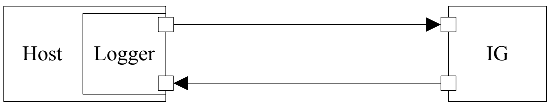


Figure C-1: Data Logger Running on a Host System

A COTS data logger will usually be run either on the host or IG and often traps any traffic at the device driver level. This data is can then be viewed at real-time or logged to a file for later viewing. In some cases traffic can be replayed.
In the case of any Ethernet based data transports a COTS package such as WireShark [6] can be used. Where other methods of transport are used (reflective memory, inter process communication etc.) there may not be a COTS solution to logging the traffic.
A generic data logger is not going to provide the automatic analysis of the CIGI frame data that we require and so is of limited use.

Redirecting Data Transport Output

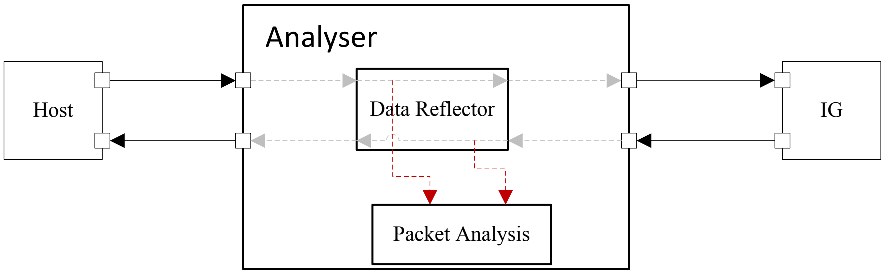


Figure C-2: Intercepting and Reflecting CIGI Data

Where supported by the data transport the data sent can be redirected to the analysis tool which will then forward the data on to its intended target before logging it. Ethernet protocols support such redirection quite easily and the analysis tool can be easily written to allow custom and exotic data transport interfaces.

This approach has the advantage of live feedback to the tester as frame data can be analysed as it is received. It is assumed that there will only be a minimal effect on the send latency of the data transport traffic because of this technique. Ideally the intercept would be done within the protocol stack (as WireShark does) to remove the additional traffic overhead but this is an implementation detail that can be left for further research.

IG COMPLIANCE TESTING

Test Methodology

The development of unit tests requires the identification of use cases and their translation into testable criteria. Use cases can be extracted from analysis of both the interface definitions in the ICD and the additional documentation supplied in the ICD (where many of the semantics of packet data are described, e.g. startup / shutdown procedure).

As well as use cases the ICD imposes a number of constraints / rules upon IG's. Where these do not fall under the notion of a use case they will be referred to as a constraint.

The testing of CIGI compliance is complicated due to a number of reasons:

- CIGI output is largely visual and cannot be automatically verified
- The specific visual representation is not specified by CIGI (i.e. interpretation of many of the packets / parameters (e.g. how a cloud looks is entirely left to the vendor) is quite valid).
- All packets, parameters and even specific scoped use of parameters are optional
- It may not be possible to translate some of the use cases identified to objectively testable criteria

Use Cases

Simple use cases can be derived from the CIGI ICD. Some (such as startup / shutdown sequence) are explicitly described in the sections prior to the packet reference. The majority need to be derived from the packets / parameters directly.

A use case should encapsulate a single specific activity that can be accomplished, i.e. as simple as possible. This is to ensure that functionality can be tested in isolation.

Each use case is defined by:

- Id
- Description
- Parameters required to support the use case
- Test setup reference
- Test steps
- ICD references
- Related packets / dependencies
- Notes

Constraints

Many of the 'shall' statements in the CIGI ICD refer to constraints placed upon the behaviour of some packets. The most notable example of this is the Start of Frame packet, behaviour regarding the IG and host frame counters, SoF being the first packet and only instance within a message are not driven by user interaction but need to be performed automatically.
There are not many IG behaviours that need to be classified this way and those that do will likely relate more to system administration than IG output.

Use Case / Constraint Fields

Each use case / constraint will require the following fields to provide enough information to produce an unambiguous test.

Id

Currently the id naming scheme has a category, major and minor numbers. Category is either a use case (U) or a constraint (C). Major numbers are incremented when starting use cases for a packet, minor numbers are incremented for each use case / constraint of a packet. E.g.

U1.3 => use case of packet 1 with unique id 3
C1.5 => constraint of packet 1 with unique id 5

Description

This should be a concise description of what is to be tested. E.g. destroy entity. It should also be as unambiguous as possible.

Test Specific Parameters

While not essential this identifies the parameters that are directly involved in the use case (as many packet parameters will not make up testable criteria within the context of the use case).

Test Setup Reference

This will contain a reference to a unique test setup procedure as defined separately (see 0).

Test Steps

This field will contain the test steps specific to demonstrating the use case, because test setups should contain the majority on setup instructions this field should not contain a large number of steps.

Pass / Fail Criteria

This should contain the conditions that determine if the IG has either passed or failed the use case. There may be a number of criteria for a use case but all should be determined from the end state of the test case (i.e. use cases do not have more than start and end states).

ICD References

This field should contain any relevant ICD references to be consulted when implementing the test or determining the validity of the test.

Dependent / Related To

While not essential this field should contain a list of packets that are required to support the test but are not directly under test, e.g. global extrapolation enable / disable requires a number of entity packets to be able to determine correct function.

Notes

This should contain any information that is considered highly relevant but perhaps not immediately obvious from the ICD or any information that the author considers important when implementing / evaluating a test.

Example Use Cases for Entity Control (V4) Packet

	A	B	C	D	E
1	Id	Use Case / Constraint Description	Test Specific Parameters	Test Setup Reference	Test Steps
2	IG Control (IGC)				
3	U1.1	Put IG into operate mode (startup)	**1.** IGC:IG Mode = 1 (Operate) **2.** SoF:IG Mode	TS1	1. Populate IGC:IG Mode = 1 (Operate). Because the IG Control and SOF packets are manditory in each data transmission there is no explicit send command needed.
4	U1.2	Put IG into standby mode (shutdown)	**1.** IGC:IG Mode = 0 (Standby/Reset) **2.** SoF:IG Mode	TS2	1. Populate an IGC:IG = 0 (Reset/Standby). Because the IG Control and SOF packets are manditory in each data transmission there is no explicit send command needed.
5	U1.3	Enable global extrapolation	**1.** Extrapolation / interpolation enable = 1	TS12	1. No further action.
6	U1.4	Disable global extrapolation	**1.** Extrapolation / interpolation enable = 0	TS12	1. Enable IGC:Smoothing Enable set to 0 (disable)

F	G	H	I
Pass / Fail Criteria	ICD Reference	Dependent / Related t	Notes
1. IG SoF return IG Mode = 1 (Operate)	4.7	Start of Frame	
1. IG SoF return IG Mode = 0 (Standby/Reset) **2.** IG is returned to default state (i.e. session state is reset)	4.7	Start of Frame	
1. Entities / attached views (with extrapolating enabled & valid velocity / acceleration) move smoothly	6.1.1 Table 5	Entity Position Entity Control Velocity Control Acceleration Control	This has no effect unless extrapolation is supported
1. Entities (with or without extrapolating enabled & valid velocity / acceleration) move only when entity position is updated (Entity Position) **2.** Attached views move only when attached entity position is updated (Entity Position) packet or view offset / attachment (View Control) is updated	6.1.1 Table 5	Entity Position Entity Control Velocity Control Acceleration Control	

Figure C-3: Example Use Case for V4.0 Entity Control Packet

Test Setups

To reduce the complexity of both test descriptions and their implementations tests will be separated into test setups that may be quite complicated and reused and test steps which should only contain those steps specific to the use case.
To reduce duplication in test setups, a test setup can reference other test setups in the setup steps.

Test Setup Fields

A test setup needs to contain an ID, description and procedure for implementing the test setup. Using references to other test setups it should be possible to write explicit and complex test setups without producing long / reproduced test setup procedures.

Id	Description	Procedure
TS1	Basic standby connection	1. The IG is initialized and is in 0 (Reset/Standby) mode. 2. The Host is running and is only sending IGC:IG Mode of 0 (Reset/Standby). 3. The inspection tool is running.
TS2	IG set to operate mode	1. TS1 2. The IG is placed in 1 (operate) mode. 3. The Host is running and is only sending IGC:IG Mode of 1 (Operate). 4. The inspection tool is running.
TS3	In synchronous mode the IG frame, and last IG frame numbers increase with each message	1. The IG and Host are running synchronously and current mode is either 0 (reset/standby) or 1 (operate mode). 2. The inspection tool is running.
TS4	In asynchronous mode, SoF last host frame numbers match range of host numbers sent since last SoF was received	1. The IG and Host are running asynchronously and current mode is either 0 (reset/standby) or 1 (operate mode). 2. The inspection tool is running.
TS5	When transitioning from operate to standby all entities are removed	1. TS2 2. Set up a View so that entities can be placed relative to the Ownship and be seen. 3. Activate several entities within the View.
TS6	When transitioning from operate to standby mission functions requests are removed and results not reported in IG messages	1. TS2 2. Request one or more mission functions every frame.

Figure C-4: Example Test Setups

Reference Terrains & Entity Models

As there is no single standard for terrain / entity model data that can be guaranteed to be correctly converted into every IG it was decided that while reference terrain data can be supplied the direct use of the reference terrains / entity models is not mandated (but is still recommended).
Where terrains and entity models are explicitly referenced by test definitions these refer to the requirements that the reference model must meet.
When describing a terrain / entity model clear and unambiguous descriptions must be used for each requirement such that it is hard to select / produce an ill-fitting substitute for the reference model. E.g. a poor requirement might read

1. the model is a 2m pyramid

While this describes the general shape of the object it is not specific enough to perform some of the tests that it may be used for (e.g. accurate entity attachment tests). A more detailed requirements definition is needed, e.g.:

1. The model is a regular square pyramid

2. The perimeter is 2m

3. The height is 2m

4. The apex is at the origin

5. The origin is the point of rotation

6. The base is at +2m above the apex (i.e. upside down)

7. The base is coloured green (0, 255, 0)

8. The north face is coloured red (255, 0, 0)

9. The east face is coloured blue (0, 0, 255)

10. The south face is coloured cyan (0, 255, 255)

11. The west face is coloured magenta (255, 0, 255)

An image of the reference model should be included with the requirements.

Example Model Reference

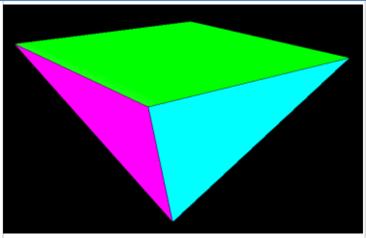
Id	Description	Reference Image	Requirements
M1	Inverted pyramid 2m		1. The model is a regular square pyramid 2. The perimeter is 2m 3. The height is 2m 4. The apex is at the origin 5. The origin is the point of rotation 6. The base is at +2m above the apex (i.e. upside down) 7. The base is coloured green (0, 255, 0) 8. The north face is coloured red (255, 0, 0) 9. The east face is coloured blue (0, 0, 255) 10. The south face is coloured cyan (0, 255, 255) 11. The west face is coloured magenta (255, 0, 255)
M2			

Figure C-5: Example Model Reference Definition

Figure C-5 shows how the model reference table should be completed for each reference model.

REPORT DOCUMENTATION PAGE

1. Recipient's Reference	2. Originator's References	3. Further Reference	4. Security Classification of Document
	STO-TR-MSG-118 AC/323(MSG-118)TP/798	ISBN 978-92-837-2123-9	PUBLIC RELEASE

5. Originator	Science and Technology Organization North Atlantic Treaty Organization BP 25, F-92201 Neuilly-sur-Seine Cedex, France
6. Title	Development of Common Image Generator Interface (CIGI) V4.0 Compliancy Testing Tools Final Report
7. Presented at/Sponsored by	Final Report of MSG-118.

8. Author(s)/Editor(s)	9. Date
Multiple	July 2018
10. Author's/Editor's Address	**11. Pages**
Multiple	96

12. Distribution Statement	There are no restrictions on the distribution of this document. Information about the availability of this and other STO unclassified publications is given on the back cover.
13. Keywords/Descriptors	CIGI Image generator Protocol

14. Abstract

Science and Technology Organization (STO) Task Group Modelling and Simulation Group (MSG)-118 “CIGI Compliancy Tools” started in October 2012 to investigate the approach and development of a method for ensuring compliance of Image Generators (IG) against the CIGI V4.0 standard SISO-STD-013-2014. As much of CIGI is optional testing compliance presented some interesting design challenges.

The task group worked for three years working from the baseline of tools provided by the United Kingdom (UK) Ministry Of Defence (MOD) under the Synthetic-natural-environments Through life cost management And Common Interfaces (STACI) research project to establish a process for defining tests and implementations of some of the identified tests.

Given significant resource issues the team managed to derive a large number of use cases and implement many tests to support compliance testing.

The output of the task group is not limited to this report but is also comprised of the test tools and content that is freely available on the opensource repository ‘SourceForge’. The Simulation Interoperability Standards Organisation (SISO) CIGI Product Support Group (PSG) has agreed to take responsibility for the administration and maintenance of the CIGI compliance tools and content on SourceForge.

NORTH ATLANTIC TREATY ORGANIZATION

SCIENCE AND TECHNOLOGY ORGANIZATION

BP 25
F-92201 NEUILLY-SUR-SEINE CEDEX • FRANCE
Télécopie 0(1)55.61.22.99 • E-mail mailbox@cso.nato.int

DIFFUSION DES PUBLICATIONS
STO NON CLASSIFIEES

Les publications de l'AGARD, de la RTO et de la STO peuvent parfois être obtenues auprès des centres nationaux de distribution indiqués ci-dessous. Si vous souhaitez recevoir toutes les publications de la STO, ou simplement celles qui concernent certains Panels, vous pouvez demander d'être inclus soit à titre personnel, soit au nom de votre organisation, sur la liste d'envoi.

Les publications de la STO, de la RTO et de l'AGARD sont également en vente auprès des agences de vente indiquées ci-dessous.

Les demandes de documents STO, RTO ou AGARD doivent comporter la dénomination « STO », « RTO » ou « AGARD » selon le cas, suivi du numéro de série. Des informations analogues, telles que le titre est la date de publication sont souhaitables.

Si vous souhaitez recevoir une notification électronique de la disponibilité des rapports de la STO au fur et à mesure de leur publication, vous pouvez consulter notre site Web (http://www.sto.nato.int/) et vous abonner à ce service.

CENTRES DE DIFFUSION NATIONAUX

ALLEMAGNE
Streitkräfteamt / Abteilung III
Fachinformationszentrum der Bundeswehr (FIZBw)
Gorch-Fock-Straße 7, D-53229 Bonn

BELGIQUE
Royal High Institute for Defence – KHID/IRSD/RHID
Management of Scientific & Technological Research for Defence, National STO Coordinator
Royal Military Academy – Campus Renaissance
Renaissancelaan 30, 1000 Bruxelles

BULGARIE
Ministry of Defence
Defence Institute "Prof. Tsvetan Lazarov"
"Tsvetan Lazarov" bul no.2
1592 Sofia

CANADA
DGSIST 2
Recherche et développement pour la défense Canada
60 Moodie Drive (7N-1-F20)
Ottawa, Ontario K1A 0K2

DANEMARK
Danish Acquisition and Logistics Organization (DALO)
Lautrupbjerg 1-5
2750 Ballerup

ESPAGNE
Área de Cooperación Internacional en I+D
SDGPLATIN (DGAM)
C/ Arturo Soria 289
28033 Madrid

ESTONIE
Estonian National Defence College
Centre for Applied Research
Riia str 12
Tartu 51013

ETATS-UNIS
Defense Technical Information Center
8725 John J. Kingman Road
Fort Belvoir, VA 22060-6218

FRANCE
O.N.E.R.A. (ISP)
29, Avenue de la Division Leclerc
BP 72
92322 Châtillon Cedex

GRECE (Correspondant)
Defence Industry & Research General Directorate, Research Directorate
Fakinos Base Camp, S.T.G. 1020
Holargos, Athens

HONGRIE
Hungarian Ministry of Defence
Development and Logistics Agency
P.O.B. 25
H-1885 Budapest

ITALIE
Ten Col Renato NARO
Capo servizio Gestione della Conoscenza
F. Baracca Military Airport "Comparto A"
Via di Centocelle, 301
00175, Rome

LUXEMBOURG
Voir Belgique

NORVEGE
Norwegian Defence Research Establishment
Attn: Biblioteket
P.O. Box 25
NO-2007 Kjeller

PAYS-BAS
Royal Netherlands Military Academy Library
P.O. Box 90.002
4800 PA Breda

POLOGNE
Centralna Biblioteka Wojskowa
ul. Ostrobramska 109
04-041 Warszawa

PORTUGAL
Estado Maior da Força Aérea
SDFA – Centro de Documentação
Alfragide
P-2720 Amadora

REPUBLIQUE TCHEQUE
Vojenský technický ústav s.p.
CZ Distribution Information Centre
Mladoboleslavská 944
PO Box 18
197 06 Praha 9

ROUMANIE
Romanian National Distribution Centre
Armaments Department
9-11, Drumul Taberei Street
Sector 6
061353 Bucharest

ROYAUME-UNI
Dstl Records Centre
Rm G02, ISAT F, Building 5
Dstl Porton Down
Salisbury SP4 0JQ

SLOVAQUIE
Akadémia ozbrojených síl gen. M.R. Štefánika, Distribučné a informačné stredisko STO
Demänová 393
031 06 Liptovský Mikuláš 6

SLOVENIE
Ministry of Defence
Central Registry for EU & NATO
Vojkova 55
1000 Ljubljana

TURQUIE
Milli Savunma Bakanlığı (MSB)
ARGE ve Teknoloji Dairesi Başkanlığı
06650 Bakanliklar – Ankara

AGENCES DE VENTE

The British Library Document Supply Centre
Boston Spa, Wetherby
West Yorkshire LS23 7BQ
ROYAUME-UNI

Canada Institute for Scientific and Technical Information (CISTI)
National Research Council Acquisitions
Montreal Road, Building M-55
Ottawa, Ontario K1A 0S2
CANADA

Les demandes de documents STO, RTO ou AGARD doivent comporter la dénomination « STO », « RTO » ou « AGARD » selon le cas, suivie du numéro de série (par exemple AGARD-AG-315). Des informations analogues, telles que le titre et la date de publication sont souhaitables. Des références bibliographiques complètes ainsi que des résumés des publications STO, RTO et AGARD figurent dans le « NTIS Publications Database » (http://www.ntis.gov).

NORTH ATLANTIC TREATY ORGANIZATION

NATO OTAN

BP 25
F-92201 NEUILLY-SUR-SEINE CEDEX • FRANCE
Télécopie 0(1)55.61.22.99 • E-mail mailbox@cso.nato.int

SCIENCE AND TECHNOLOGY ORGANIZATION

DISTRIBUTION OF UNCLASSIFIED STO PUBLICATIONS

AGARD, RTO & STO publications are sometimes available from the National Distribution Centres listed below. If you wish to receive all STO reports, or just those relating to one or more specific STO Panels, they may be willing to include you (or your Organisation) in their distribution.

STO, RTO and AGARD reports may also be purchased from the Sales Agencies listed below.

Requests for STO, RTO or AGARD documents should include the word 'STO', 'RTO' or 'AGARD', as appropriate, followed by the serial number. Collateral information such as title and publication date is desirable.

If you wish to receive electronic notification of STO reports as they are published, please visit our website (http://www.sto.nato.int/) from where you can register for this service.

NATIONAL DISTRIBUTION CENTRES

BELGIUM
Royal High Institute for Defence – KHID/IRSD/RHID
Management of Scientific & Technological Research for Defence, National STO Coordinator
Royal Military Academy – Campus Renaissance
Renaissancelaan 30
1000 Brussels

BULGARIA
Ministry of Defence
Defence Institute "Prof. Tsvetan Lazarov"
"Tsvetan Lazarov" bul no.2
1592 Sofia

CANADA
DSTKIM 2
Defence Research and Development Canada
60 Moodie Drive (7N-1-F20)
Ottawa, Ontario K1A 0K2

CZECH REPUBLIC
Vojenský technický ústav s.p.
CZ Distribution Information Centre
Mladoboleslavská 944
PO Box 18
197 06 Praha 9

DENMARK
Danish Acquisition and Logistics Organization (DALO)
Lautrupbjerg 1-5
2750 Ballerup

ESTONIA
Estonian National Defence College
Centre for Applied Research
Riia str 12
Tartu 51013

FRANCE
O.N.E.R.A. (ISP)
29, Avenue de la Division Leclerc – BP 72
92322 Châtillon Cedex

GERMANY
Streitkräfteamt / Abteilung III
Fachinformationszentrum der Bundeswehr (FIZBw)
Gorch-Fock-Straße 7
D-53229 Bonn

GREECE (Point of Contact)
Defence Industry & Research General Directorate, Research Directorate
Fakinos Base Camp, S.T.G. 1020
Holargos, Athens

HUNGARY
Hungarian Ministry of Defence
Development and Logistics Agency
P.O.B. 25
H-1885 Budapest

ITALY
Ten Col Renato NARO
Capo servizio Gestione della Conoscenza
F. Baracca Military Airport "Comparto A"
Via di Centocelle, 301
00175, Rome

LUXEMBOURG
See Belgium

NETHERLANDS
Royal Netherlands Military Academy Library
P.O. Box 90.002
4800 PA Breda

NORWAY
Norwegian Defence Research Establishment, Attn: Biblioteket
P.O. Box 25
NO-2007 Kjeller

POLAND
Centralna Biblioteka Wojskowa
ul. Ostrobramska 109
04-041 Warszawa

PORTUGAL
Estado Maior da Força Aérea
SDFA – Centro de Documentação
Alfragide
P-2720 Amadora

ROMANIA
Romanian National Distribution Centre
Armaments Department
9-11, Drumul Taberei Street
Sector 6
061353 Bucharest

SLOVAKIA
Akadémia ozbrojených síl gen M.R. Štefánika, Distribučné a informačné stredisko STO
Demänová 393
031 06 Liptovský Mikuláš 6

SLOVENIA
Ministry of Defence
Central Registry for EU & NATO
Vojkova 55
1000 Ljubljana

SPAIN
Área de Cooperación Internacional en I+D
SDGPLATIN (DGAM)
C/ Arturo Soria 289
28033 Madrid

TURKEY
Milli Savunma Bakanlığı (MSB)
ARGE ve Teknoloji Dairesi Başkanlığı
06650 Bakanliklar – Ankara

UNITED KINGDOM
Dstl Records Centre
Rm G02, ISAT F, Building 5
Dstl Porton Down, Salisbury SP4 0JQ

UNITED STATES
Defense Technical Information Center
8725 John J. Kingman Road
Fort Belvoir, VA 22060-6218

SALES AGENCIES

The British Library Document Supply Centre
Boston Spa, Wetherby
West Yorkshire LS23 7BQ
UNITED KINGDOM

Canada Institute for Scientific and Technical Information (CISTI)
National Research Council Acquisitions
Montreal Road, Building M-55
Ottawa, Ontario K1A 0S2
CANADA

Requests for STO, RTO or AGARD documents should include the word 'STO', 'RTO' or 'AGARD', as appropriate, followed by the serial number (for example AGARD-AG-315). Collateral information such as title and publication date is desirable. Full bibliographical references and abstracts of STO, RTO and AGARD publications are given in "NTIS Publications Database" (http://www.ntis.gov).

ISBN 978-92-837-2123-9