



**Simulation Interoperability  
Standards Organization**

*"Simulation Interoperability & Reuse through Standards"*

## **2022 Virtual SIW**

### **LIVE Tutorials – Monday 7 February**

#### **0800-1200 (EST)**

##### **Employing the C2-Simulation Interoperation (C2SIM) Standard for Coalition Military Operations and Exercises (*Part One*)**

Technical Course MSG-194, organized by the NATO Modelling and Simulation Group

**Attendees MUST register for this two-part course through the NATO STO site:**

Event Detailed List - MSG-194 Employing the C2-Simulation Interoperation (C2SIM) Standard for Coalition Military Operations and Exercises ([nato.int](https://nato.int))

Part Two of this Technical Course will be held on **Friday 11 February, 0800 to 1200**.

Course Agenda: [Copy of Programme MSG-194 TC \(\[nato.int\]\(https://nato.int\)\)](#)

#### **1030-1200 (EST)**

##### **Distributed Interactive Simulation (DIS) 201 - New Extensibility and Dead Reckoning Features in DIS Version 7 and 8**

*Instructor: **Bob Murray**, SimPhonics*

**Join Meeting:** <https://meet.goto.com/SISO/tutorials-one>

**Access Code:** 886-722-725

*Phone Number if needed: (571) 317-3116*

*(See international numbers at end)*

This tutorial provides an in-depth review of two features: PDU extensibility and improvements in dead reckoning.

PDU extensibility expands the ability of DIS users to add custom data to PDUs. Some PDUs allow user-defined records to be directly added. Other PDUs can be extended using the Attribute PDU in DIS Version 7. Both methods retain compatibility with older versions of DIS. This allows customized PDUs to be added in new or upgraded simulations while maintaining interoperability to older simulations that cannot be modified. The upcoming DIS Version 8 proposes to add extensibility directly to every PDU but will not maintain forward compatibility.



**Simulation Interoperability  
Standards Organization**

*"Simulation Interoperability & Reuse through Standards"*

Dead reckoning has been enhanced in DIS Version 7, mainly in the extrapolation of entity orientation. A new geometric method of determining the orientation threshold is described using either quaternions or rotation matrices. This method avoids the problems of Euler angle singularities that can cause excessively high PDU transmit rates. DIS Version 8 proposes a new dead reckoning algorithm. It performs as well or better than the traditional algorithms from older DIS versions and is easier to use.

### **1230-1400 (EST)**

**(1230-1300) 2022-SIW-19** - Autogenerating and Implementing the open-dis7-java Codebase for Distributed Interactive Simulation (DIS) Protocol Streaming

**(1300-1400) Tutorial:** Using open-dis7 Java and Python Codebases for Distributed Interactive Simulation (DIS) Protocol Streaming

*Presenter/Instructor: **Don Brutzman**, Naval Postgraduate School (NPS)*

**Join Meeting:** <https://meet.goto.com/SISO/tutorials-one>

**Access Code:** 886-722-725

*Phone Number if needed: (571) 317-3116*

*(See international numbers at end)*

**2022-SIW-19** - Autogenerating and Implementing the open-dis7-java Codebase for Distributed Interactive Simulation (DIS) Protocol Streaming

A distinguishing hallmark of the IEEE Distributed Interactive Simulation (DIS) Protocol is that it provides data-centric representations for carefully defined data messages. Message structures, semantics and interaction patterns have only been standardized following extensive working-group evaluation. Implementing software that produces and parses the strict data messages conforming to this open standard is permitted to vary widely. The Open-DIS project is a long-running effort to produce open-source software libraries that implement the DIS protocol in a variety of programming languages, unrestricted for any use. Motivations for use include modeling of realistic representations of entities and behaviors, distributed simulation programming, and practical interoperation between applications to promote collaboration, research and education. An innovative design represented data structures for all DIS Protocol Data Units (PDUs) in XML, permitting source-code generators to produce a variety of functionally similar codebase libraries. The original project provides support for approximately half of the DIS vocabulary using multiple programming languages (Java, Python, JavaScript, C++/C#) and file encodings (XML, JSON, EXI). Expanding software usage and adaptation continue to spur improvements in best practices.



## Simulation Interoperability Standards Organization

*"Simulation Interoperability & Reuse through Standards"*

The open-dis7-java library is a major upgrade now providing 100% coverage of all 72 PDU types in IEEE DIS version 7, as well as full coverage of over 22,000 SISO Enumeration data structures identifying diverse entities, sensors, weapons, domains, nationalities, etc. as Plain Old Java Objects (POJOs). A singleton threaded network interface class facilitates integration of DIS reading and writing together with diverse Java applications. Special emphasis has been placed on recording of PDU streams in multiple encodings (native binary, plaintext, base64, XML, EXI, JSON). Following the principle "a stream is a stream" we are beginning to show support for repeatable unit testing with expected benefits for sustainable Live Virtual Constructive (LVC) interoperability, Validation Verification Accreditation (VV+A), and Testing Development Operations (TestDevOps). A growing set of examples, course projects, and tutorial assets demonstrate effective open-dis7-java library usage.

We have begun to transition academic development, issue tracking and code improvements back into public forums on GitHub. Work in progress includes design refinement and autogeneration of an open-dis7-python library, as well as track distillation to create both X3D animation interpolators plus corresponding KML waypoints for track visualization. Planned work for 2022 includes future support for Compressed-DIS (C-DIS) and DISv8 protocols, plus continued adaptation of Rich Semantic Track (RST) principles as part of an Unmanned Systems Data Strategy. A unified approach to bridging DIS streams across C2SIM, TENA, and HLA RPR FOM environments has potential to further demonstrate broad LVC interoperability. Such repetition of possible, planned, rehearsed, actual, replayed, and analyzed streams can provide the missing links needed for effectively informing real-world operations. We expect that continued exploration of real, virtual and hybrid exercise streaming can establish a common path for Modeling and Simulation (M+S), Command and Control (C2), and unmanned-systems experimentation to be actionable as repeatable Big Data across all domains.

### **Tutorial:** Using open-dis7 Java and Python Codebases for Distributed Interactive Simulation (DIS) Protocol Streaming

The Distributed Interactive Simulation (DIS) protocol is a well-established IEEE standard for packet-level exchange of state information between entities in military simulations. DIS facilitates simulation interoperability through a consistent over-the-wire format for information, widely agreed upon constant enumeration values, and community-consensus semantics. Anyone can obtain the IEEE-1278 standard and implement their own compliant, interoperable, DIS application. A large variety of tools and codebases simplify this effort, enabling multi-architecture integration of simulations using the DIS stand baseline. DIS focus begins with real-time, physics-based, entity-scale simulations, providing state update and interaction mechanisms which can scale to large virtual environments.

This tutorial is a "DIS 202" exploration for software implementers and simulation-systems integrators. Examples are provided using the open-source open-dis7-java library for DIS v7 and Enumerations support, available primarily in Java with further support planned for multiple programming languages. This work first demonstrates the source-code autogeneration process



## Simulation Interoperability Standards Organization

*"Simulation Interoperability & Reuse through Standards"*

that turns XML-based DIS PDU representations into code libraries demonstrating best practices. Numerous examples for application testing and course work are examined. Ongoing work includes unit testing and analysis of DIS streams, potential codebase production for Compressed DIS and DISv8 standards, and Web-based implementations using X3D Graphics.

\*\*\*\*\*

### **International phone numbers:**

Australia: [+61 2 9091 7603](tel:+61290917603)  
Austria: [+43 7 2081 5337](tel:+43720815337)  
Belgium: [+32 28 93 7002](tel:+3228937002)  
Bulgaria: [+359 2 906 0606](tel:+35929060606)  
Canada: [+1 \(647\) 497-9373](tel:+16474979373)  
Czech Republic: [+420 2 96 21 62 28](tel:+420296216228)  
Denmark: [+45 32 72 03 69](tel:+4532720369)  
Finland: [+358 923 17 0556](tel:+358923170556)  
France: [+33 187 210 241](tel:+33187210241)  
Germany: [+49 721 6059 6510](tel:+4972160596510)  
Greece: [+30 21 0 300 2693](tel:+302103002693)  
Hungary: [+36 1 933 3700](tel:+3619333700)  
Ireland: [+353 15 360 756](tel:+35315360756)  
Israel: [+972 3 376 3071](tel:+97233763071)  
Italy: [+39 0 230 57 81 80](tel:+390230578180)  
Netherlands: [+31 202 251 001](tel:+31202251001)  
New Zealand: [+64 9 282 9510](tel:+6492829510)  
Norway: [+47 21 93 37 37](tel:+4721933737)  
Spain: [+34 932 75 1230](tel:+34932751230)  
Sweden: [+46 853 527 818](tel:+46853527818)  
Switzerland: [+41 225 4599 60](tel:+41225459960)  
Turkey: [+90 212 900 4812](tel:+902129004812)  
United Kingdom: [+44 20 3713 5011](tel:+442037135011)