Virtual Distributed Interactive Simulation (VDIS) Specification

for

**Synthetic Environment (SE) Core**

**Common Virtual Environment Management (CVEM)**

Contract No.: W900KK-11-D-0006

Document No.: SECORE-CVEM-U-0000303

Revision:

June 19, 2015

**Prepared by:**Leidos, Inc.  
Surveillance & Reconnaissance Group  
12901 Science Drive  
Orlando, Florida 32826

Revision History

| Revision | Date | Description |
| --- | --- | --- |
| 0.1 | 27 Feb 2007 | Initial Draft |
| 0.2 | 02 May 2007 | Revised draft available for PM CATT review prior to first WG |
| 0.6 | 06 June 2007 | Partial VDIS specification draft available following program analysis and consolidation; including initialization, minefields, weather, network protocols, datum records, updated Stealth, etc. |
| 0.8 | 29 June 2007 | VDIS specification draft available for Build 3.3. Revised entity types enumeration list, added quite a few enumerations. |
| 1.0 | 15 October 2007 | Split out “future” topics to clarify what parts are really supported in the first official release (1.0). Add support for civilian vehicles, life forms, and portable weapons. Revised/finalized datums. |
| 1.1 | 19 February 2009 | EE PDU, digital messaging, laser designator, extended appearance fields, SIMAN and health status, Jammers, Stealth, Attribute PDU, Application Control, Sling Loads, PDU Header changes, protocol version rules, Exercise IDs, enumerations, rules on identifiers, VP records, rules on identifiers, Collision, life form fields, high fidelity thermal, and entity association. |
| 1.2 | 28 October 2009 | Clarified Entity Offset VP record and the Station Location, clarified action responses from sling loads, updated Action ID for Sling Load Capability Request (to 4300), Repair – no Data Request PDU (typo), revised PDU Type (not simply supported, but all PDU Types), removed PDU Family 129 and all of its PDUs, added timer for sling load requests, removed Beam Motion Pattern (it’s now in the Attribute record and not currently supported) |
| 1.3 | 05 January 2010 | Added section for Tactical Smoke (VDIS-61), added clarification paragraph to Force ID (VDIS-86), revised digital messaging (VDIS-87 and VDIS-97), revised sling loads for further clarification (VDIS-96), revised Time Management (VDIS-112). |
| 1.3.1 | 08 February 2010 | Identified units of measure for liquids (fuels, etc.) (VDIS-105), deleted second sentence of 12.2 (6) (VDIS-122). Changed Distribution D statement to Distribution A. |
| 1.4 | 4 June 2010 | Removed Antenna Location VP record (VDIS-155), added description of “Number of Info Messages” field (VDIS-118), added Legacy Extended Life Form Appearance enumeration to Table 2.6 (VDIS-131), added VP records for extended appearance on supplies and cultural features (VDIS-145), revised Task Organization and Entity Marking sections and removed all other references to the task organization or chevron marking VP records (VDIS-158), added appendix B and C to VDIS Spec (VDIS-102), updated minefield lane marker and breach (VDIS-144), weather record clarification (VDIS-143), clarified environment type (VDIS-136), clarify color usage (VDIS-159), updated laser designator (VDIS-129). Table 1.2 brought up to date (VDIS-130), Added entity types for RVS gaps (VDIS-177), Section 3.26 Life Form Parachuting (VDIS-135). |
| 1.5 | 21 July 2010 | Specified bit assertion values for all “On” and “Off” enumerants (VDIS-156). Changed specification for identifying minefield breach (VDIS-181). Changed specification for numbering convention used to identify individual designator systems in the Designator PDU (VDIS-173). Specified representation for Point and Linear Object Breaching (VDIS-187) |
| 1.5.1 | 21 September 2010 | Add extended definition for the Launcher bit in the entity appearance record (VDIS-190). Add remarks in Entity Marking and Force ID sections stating how Force ID is used to in conjunction with Entity Marking (VDIS-191). |
| 1.5.2 | 15 November 2010 | Revise definition for the Launcher bit in the entity appearance record (VDIS-196). Add direction for how to fill in FAD/# when not supplied in the tactical message (intended for certain ACK messages) (VDIS-195). |
| 1.5.3 | 4 January 2011 | Update version to 1.5.3. Added Minefield Entry and Mine Detonation (VDIS-201). Added 3 entity types to support NCM3 and required appearance bits capabilities for doors (VDIS-204). Added Hoisting section (VDIS-203). Added “sling load pendant” to life forms (VDIS-205). Clarified usage of Event Report datums (VDIS-202). |
| 1.5.4 | 03 February 2011 | Add M1151 (VDIS-206), and document here the articulated parts mappings in new Appendix B. |
| 1.5.5 | 08 February 2011 | Update RVS entities to use articulated parts (VDIS-207), document in the Appendix B. |
| 1.5.6 | 18 February 2011 | Add JIEDDO support (VDIS-208), enumeration updates only. |
| 1.5.7 | 1 June 2011 | Revise/update event report issues (VDIS-214), provide JIEDDO threshold support changes (VDIS-216), clarify AVLB articulated parts (VDIS-218), update text to draft 15 and remove unneeded duplicate or unused text (VDIS-220). |
| 1.6 | 22 July 2011 | Update objects for JIEDDO support (VDIS-221), clarification for CFS (VDIS-199). |
| 1.6.1 | 09 December 2011 | Remove Section 3.6.3 Surface Platforms (VDIS-108), support for multiple colors (VDIS-200), PCR 207B Extended Platform Appearance revision (VDIS-100), PCR 215B Extended Life Form Appearance revision (VDIS-115), PCR 211B Sling Loads (VDIS-103), PCR 209A Entity Offset VP (VDIS-146), PCR 210A Dead Reckoning VP Record (VDIS-147), PCR 214B IEDs (VDIS-149), PCR 218C Dynamic Environment (VDIS-150), PCR 216B Mounting/Towing (VDIS-174), PCR 208B Application Control (VDIS-104, includes a clarification to Exercise ID per the PCR), M1200 (VDIS-222), Radio Category (VDIS-110, changes to XML only), PCR 212 Weather (VDIS-212, changes to Appendix A). |
| SE Core CVEM Contract Release | | |
| A | 28 February 2013 | Revise Extended Air Platform Equipment Appearance record to indicate the active status of the ATIRCM (734), Extended Air Platform Appearance for AVCATT (774), Step VDIS up to 1278.1-2012 (1071), Remove various unused capabilities (1123, 1149, 1150, 1151, 1152, 1153), Support CCTT Concurrency (1161).  Changes in enumerations only: Object Types for MEL (758), Entity Types from MEL (770), AVCATT Entity Types (772), CM2 Lot 13.1 tuples (1065), Clean Up Enumerations in VDIS (1069) |
| B | 29 March 2013 | Coordinate Systems in VDIS (1458).  Changes in enumerations only: CM2 Lot 13.3 tuples (1249), Hellfire enumerations (1459). |
| C | 07 June 2013 | Remove unused sections: Voice Communications (8.4), Intercom (8.5), Information Operations (8.6), Exercise Layout (11) (1457); support for hidden entities with present domain (1878); EDM updates for crater and trees appearance and object types (1872).  Changes in enumerations only: CM2 Lot 13.5 tuples (1901), Missing MEL tuples (1914). |
| D | 07 August 2013 | Alignment with SISO enumerations for Entity Types (2040), CM2 Lot 13.7 (2033), Hydra Expendables (2036), Surface Appearance (2037), Dead Animals (2039). |
| E | 07 October 2013 | CM2 Lot 13.9 (2034), Ammo pallets (2134), entity marking (1810), entity offset VP record clarification (2138), SISO alignment (2136) |
| F | 06 December 2013 | CM2 Lot 13.11 (2035), Named Static Geo Filter (2209), Spelling errors in VDIS Enums (2348), Cleanup Entity/Obj Appearance (2373), Tent Aperture (2369), Mokopa ATGM (2421), Missing MEL types (2551) |
| G | 04 April 2014 | AH-64E without Longbow (2584), CM2 Lot 1for 2014 (2998), Add missing MEL types (2814), Mounted enums (2968), M3A3 ODS SA (3060), Add missing UUIDs (3068), LBAND SATCOM Radio System and CAT (3086) |
| H | 06 June 2014 | Cleanup (3444), CM2 Lot 14.05 (3441), ARNG Lots 1 and 2 (3405), Updates for HiFi JP and Fast Rope (3180), Add JP and Sked (3179) |
| I | 07 November 2014 | (3484) Applicability Clarification to Object Appearance, (3658) Hellfire AGM-114 Additions, (4101) Add remaining ARNG Entities, (4103) Update HiFi Hoisting, (4104) Add Bambi Bucket, (4105) Add Articulated Parts for Search Lights, (4552) Appearance Table UIDs and Applicability, (4648) Resolve Duplicate UUIDs and SEPTUPLES, (4681) ASE Upgrade Entities |
| J | 13 February 2015 | AVCATT RA (5405), Tripod launchers (4975), Radar Perception Record (4917), ARNG Items (4938), Forest fires and floods (4939), Skyguard Missile Launcher (4974) |
| K | 03 April 2015 | Debris (5427), current template (5439), SA-24 Strelets (5461), M3A# CMED BFIST (5462), Insurgent building colors (5494) |
| L | 19 June 2015 | Support CEVT Requirements (2861) |

Table of Contents

[List of Figures ix](#_Toc422237143)

[List of Tables ix](#_Toc422237144)

[1 INTRODUCTION 1](#_Toc422237145)

[1.1 Purpose 1](#_Toc422237146)

[1.2 Scope 1](#_Toc422237147)

[2 REFERENCE DOCUMENTS 1](#_Toc422237148)

[3 GENERAL REQUIREMENTS 2](#_Toc422237149)

[3.1 DIS Protocol Data Units (PDUs) 2](#_Toc422237150)

[3.1.1 Protocol Data Units 2](#_Toc422237151)

[3.1.2 PDU Header 3](#_Toc422237152)

[3.1.3 PDU Usage, Issuance, and Receipt Rules 3](#_Toc422237153)

[3.1.4 Heartbeats and Timeouts 4](#_Toc422237154)

[3.2 Variable Parameter (VP) Records 4](#_Toc422237155)

[3.3 Coordinate Systems 4](#_Toc422237156)

[3.4 Time Management 4](#_Toc422237157)

[3.5 Dead Reckoning Algorithm (DRA) 5](#_Toc422237158)

[4 ENTITIES 5](#_Toc422237159)

[4.1 Entity Creation / Removal 5](#_Toc422237160)

[4.2 Entity Identifier Record 5](#_Toc422237161)

[4.3 Force ID 5](#_Toc422237162)

[4.4 Entity Type and Alternate Entity Type 6](#_Toc422237163)

[4.5 Entity Appearance 6](#_Toc422237164)

[4.5.1 Land Platforms 6](#_Toc422237165)

[4.5.2 Air Platforms 7](#_Toc422237166)

[4.5.3 Other Platforms and Munitions 7](#_Toc422237167)

[4.5.4 Life Forms 7](#_Toc422237168)

[4.5.5 Environmental, Cultural Features, and Supplies 7](#_Toc422237169)

[4.5.6 Radios, Expendables, and Sensor/Emitters 7](#_Toc422237170)

[4.6 Entity Marking 7](#_Toc422237171)

[4.7 Capabilities 8](#_Toc422237172)

[4.8 Life Forms 8](#_Toc422237173)

[4.8.1 Animal Life Forms 9](#_Toc422237174)

[4.8.2 Dead Animals 10](#_Toc422237175)

[4.8.3 Head Gazing / Weapon Aiming 10](#_Toc422237176)

[4.8.4 Action Sequences 12](#_Toc422237177)

[4.8.5 Legacy Extended Life Form Appearance VP Record 14](#_Toc422237178)

[4.9 Platforms 16](#_Toc422237179)

[4.9.1 Extended Platform Appearance VP Record 16](#_Toc422237180)

[4.9.2 Civilian Vehicles 19](#_Toc422237181)

[4.10 Extended Appearance for Other Entities 20](#_Toc422237182)

[4.10.1 Extended Cultural Feature Appearance VP Record 20](#_Toc422237183)

[4.10.2 Extended Supply Appearance VP Record 22](#_Toc422237184)

[4.11 Present Domain 23](#_Toc422237185)

[4.12 High Fidelity Lights 23](#_Toc422237186)

[4.13 High Fidelity Thermal Sensing 25](#_Toc422237187)

[4.14 Mounting / Towing / Hoisting 26](#_Toc422237188)

[4.14.1 Mounting 26](#_Toc422237189)

[4.14.2 Towing 29](#_Toc422237190)

[4.14.3 Hoisting 31](#_Toc422237191)

[4.15 Associated Entities 34](#_Toc422237192)

[4.15.1 Entity Association VP Record 34](#_Toc422237193)

[4.15.2 Entity Offset VP record 35](#_Toc422237194)

[4.15.3 Dead Reckoning VP record 37](#_Toc422237195)

[4.16 Weapons and Detonations 38](#_Toc422237196)

[4.16.1 Weapons Fire 38](#_Toc422237197)

[4.16.2 Detonations 38](#_Toc422237198)

[4.17 Collisions 38](#_Toc422237199)

[4.18 Entity Status 39](#_Toc422237200)

[4.18.1 PDU Usage 39](#_Toc422237201)

[4.18.2 Entity Status Datum IDs 40](#_Toc422237202)

[4.19 Sling Loads 40](#_Toc422237203)

[4.19.1 Introduction 40](#_Toc422237204)

[4.19.2 General Requirements 40](#_Toc422237205)

[4.19.3 Capability Requests 41](#_Toc422237206)

[4.19.4 Attachment 41](#_Toc422237207)

[4.19.5 Detachment 43](#_Toc422237208)

[4.20 Life Form Parachuting 44](#_Toc422237209)

[4.20.1 Visualization 44](#_Toc422237210)

[4.20.2 Mounted on Aircraft 44](#_Toc422237211)

[4.20.3 Freefall Descent (OPTIONAL) 44](#_Toc422237212)

[4.20.4 Deployed Descent 44](#_Toc422237213)

[4.20.5 First Surface Contact 45](#_Toc422237214)

[4.20.6 Chute Detached 45](#_Toc422237215)

[4.20.7 Known issues 45](#_Toc422237216)

[5 SIMAN 45](#_Toc422237217)

[5.1 Reliability 45](#_Toc422237218)

[5.1.1 Action Request / Action Response 46](#_Toc422237219)

[5.1.2 Data Query / Set Data / Data 46](#_Toc422237220)

[5.2 Application Control PDU 47](#_Toc422237221)

[5.2.1 Purpose 47](#_Toc422237222)

[5.2.2 Information contained in the Application Control PDU 47](#_Toc422237223)

[5.2.3 Issuance of the Application Control PDU 47](#_Toc422237224)

[5.2.4 Receipt of the Application Control PDU 48](#_Toc422237225)

[5.2.5 PDU Definition 49](#_Toc422237226)

[5.2.6 Application Control Enumerations 50](#_Toc422237227)

[5.3 Scenario Initialization 51](#_Toc422237228)

[5.3.1 Initialization Files 51](#_Toc422237229)

[5.3.2 Initialization Sequence 51](#_Toc422237230)

[5.4 Asset Control 53](#_Toc422237231)

[5.4.1 Asset Discovery 53](#_Toc422237232)

[5.4.2 Application State and Capability 53](#_Toc422237233)

[5.4.3 Application Health Status 54](#_Toc422237234)

[5.4.4 Asset Resignation 55](#_Toc422237235)

[5.4.5 Daily Readiness Check 55](#_Toc422237236)

[5.5 Exercise Management 55](#_Toc422237237)

[5.5.1 Join Exercise 55](#_Toc422237238)

[5.5.2 Resign from Exercise 56](#_Toc422237239)

[5.5.3 Start / Resume 57](#_Toc422237240)

[5.5.4 Stop / Freeze 58](#_Toc422237241)

[5.5.5 Restart 59](#_Toc422237242)

[5.5.6 Reconstitute 59](#_Toc422237243)

[5.5.7 Reposition 59](#_Toc422237244)

[5.5.8 Checkpoint / Reset 59](#_Toc422237245)

[5.5.9 Exercise Management 59](#_Toc422237246)

[5.5.10 Event Reporting 59](#_Toc422237247)

[5.5.11 Comments 61](#_Toc422237248)

[6 STEALTH 61](#_Toc422237249)

[6.1 Stealth State 61](#_Toc422237250)

[6.2 Stealth Control 62](#_Toc422237251)

[7 NETWORK PROTOCOLS 62](#_Toc422237252)

[7.1 Network Transport Layer Rules 62](#_Toc422237253)

[7.1.1 Key Assumptions 62](#_Toc422237254)

[7.1.2 Network Packet Filtering Approaches 63](#_Toc422237255)

[7.1.3 PDU Bundling 64](#_Toc422237256)

[7.2 Performance Requirements 64](#_Toc422237257)

[7.3 Error Detection 64](#_Toc422237258)

[8 TASK ORGANIZATION 64](#_Toc422237259)

[8.1 Aggregate State PDU for Task Organization 65](#_Toc422237260)

[8.2 Command from Simulator (CFS) 66](#_Toc422237261)

[8.2.1 Description 66](#_Toc422237262)

[8.2.2 Initialization 66](#_Toc422237263)

[8.2.3 Task Reorganization 66](#_Toc422237264)

[9 COMMUNICATIONS 67](#_Toc422237265)

[9.1 Transmitter PDU 67](#_Toc422237266)

[9.2 Signal PDU 67](#_Toc422237267)

[9.3 Tactical Data Link (TDL) 67](#_Toc422237268)

[9.3.1 Modulation Parameters of Transmitter PDU 67](#_Toc422237269)

[9.3.2 Link 16 68](#_Toc422237270)

[9.3.3 Link 11/11B 68](#_Toc422237271)

[9.3.4 EPLRS / SINCGARS / L-Band Satcom 68](#_Toc422237272)

[10 LOGISTICS 73](#_Toc422237273)

[10.1 Repair 73](#_Toc422237274)

[10.2 Recovery 74](#_Toc422237275)

[10.3 Resupply 74](#_Toc422237276)

[10.4 Future Capabilities 74](#_Toc422237277)

[11 DYNAMIC ENVIRONMENT 75](#_Toc422237278)

[11.1 Dynamic Environment Objects 75](#_Toc422237279)

[11.1.1 Point Object State PDU 77](#_Toc422237280)

[11.1.2 Linear Object State PDU 77](#_Toc422237281)

[11.1.3 Areal Object State PDU 77](#_Toc422237282)

[11.2 Weather 77](#_Toc422237283)

[11.2.1 Weather Manager 77](#_Toc422237284)

[11.2.2 Global and Regional Weather Identification 78](#_Toc422237285)

[11.2.3 Weather Records 78](#_Toc422237286)

[11.2.4 Weather Environment Type 79](#_Toc422237287)

[11.3 Minefields 80](#_Toc422237288)

[11.3.1 Areal Object State 80](#_Toc422237289)

[11.3.2 Minefield Breaching 80](#_Toc422237290)

[11.4 Improvised Explosive Devices (IEDs) 80](#_Toc422237291)

[11.4.1 Object State General Appearance 81](#_Toc422237292)

[11.4.2 Object/Entity Association record 81](#_Toc422237293)

[11.5 Tactical Smoke 82](#_Toc422237294)

[11.6 Breaching Point and Linear Objects 82](#_Toc422237295)

[11.7 Dynamic Terrain 82](#_Toc422237296)

[11.7.1 Scaled Height Map PDU 83](#_Toc422237297)

[11.7.2 Soil Attribute Map PDU 88](#_Toc422237298)

[12 DISTRIBUTED EMISSION REGENERATION (DER) 95](#_Toc422237299)

[12.1 Electromagnetic Emission 95](#_Toc422237300)

[12.1.1 Jamming Technique 96](#_Toc422237301)

[12.1.2 Electromagnetic Emission PDU Issuance Rules 96](#_Toc422237302)

[12.1.3 Electromagnetic Emission PDU Receipt Rules 96](#_Toc422237303)

[12.2 Designator 96](#_Toc422237304)

[12.2.1 Designator PDU Enumerations 98](#_Toc422237305)

[12.2.2 Flashing Lasers and Power Modes 99](#_Toc422237306)

[12.3 Supplemental Emission / Entity State (SEES) 100](#_Toc422237307)

[12.4 Identification, Friend, or Foe (IFF) 100](#_Toc422237308)

[APPENDIX A VARIABLE RECORDS 101](#_Toc422237309)

[APPENDIX B ENTITY SPECIFIC INFORMATION 102](#_Toc422237310)

[APPENDIX C KEY TERMS AND ACRONYMS 106](#_Toc422237311)

List of Figures

[Figure 4‑1. VDIS Hoisting 34](#_Toc422237312)

[Figure 5‑1. VDIS Scenario Initialization 52](#_Toc422237313)

[Figure 5‑2. VDIS Application State and Capability 54](#_Toc422237314)

[Figure 5‑3. VDIS Application Health Status 55](#_Toc422237315)

[Figure 5‑4. VDIS Join Exercise 56](#_Toc422237316)

[Figure 5‑5. VDIS Resign from Exercise 57](#_Toc422237317)

[Figure 5‑6. VDIS Start/Resume 58](#_Toc422237318)

[Figure 5‑7. VDIS Stop/Freeze 59](#_Toc422237319)

[Figure 11‑1. Weather Manager 78](#_Toc422237320)

[Figure 11‑3. Soil Attribute Map PDU Fields 88](#_Toc422237321)

[Figure 11‑4. Example of a Simple Grid (not to scale) 90](#_Toc422237322)

List of Tables

[Table 2‑1. Reference Documents 1](#_Toc422237323)

[Table 3‑1. Variable Parameters Record Type Enumeration 4](#_Toc422237324)

[Table 4‑1. Entity Capabilities 8](#_Toc422237325)

[Table 4‑2. Head Gazing / Weapon Aiming VP Record 11](#_Toc422237326)

[Table 4‑3. Life Form Action Sequence VP Record 13](#_Toc422237327)

[Table 4‑4. Legacy Extended Life Form Appearance VP Record 15](#_Toc422237328)

[Table 4‑5. Extended Platform Appearance VP Record 17](#_Toc422237329)

[Table 4‑6. Extended Cultural Feature Appearance VP Record 21](#_Toc422237330)

[Table 4‑7. Extended Supply Appearance VP Record 22](#_Toc422237331)

[Table 4‑8. Present Domain Enumeration 23](#_Toc422237332)

[Table 4‑9. High Fidelity Lights VP Record – Values 24](#_Toc422237333)

[Table 4‑10. High Fidelity Thermal VP Record 26](#_Toc422237334)

[Table 4‑11. Entity Association Example (RWA) 35](#_Toc422237335)

[Table 4‑12. Entity Offset VP Record 36](#_Toc422237336)

[Table 4‑13. Dead Reckoning VP Record 37](#_Toc422237337)

[Table 4‑14. Collision Origin Enumeration of the Collision PDU 39](#_Toc422237338)

[Table 4‑15. Entity Status Datums 40](#_Toc422237339)

[Table 5‑1. Comparison of Reliability PDUs 46](#_Toc422237340)

[Table 5‑2. Application Control PDU 49](#_Toc422237341)

[Table 5‑3. Control Type Enumeration for Application Control PDU 50](#_Toc422237342)

[Table 5‑5. AAR Event Report Datum IDs 60](#_Toc422237343)

[Table 5‑6. Damage Cause Enumeration 61](#_Toc422237344)

[Table 5‑7. Damage Extent Enumeration 61](#_Toc422237345)

[Table 5‑8. Mission Type Enumeration 61](#_Toc422237346)

[Table 5‑9. Radar Track Status Enumeration 61](#_Toc422237347)

[Table 9‑1. VDIS Digital Message 72](#_Toc422237348)

[Table 9‑2. Signal PDU with VDIS Digital Messages 73](#_Toc422237349)

[Table 11‑1. Terrain Feature Characterization 75](#_Toc422237350)

[Table 11‑2. Example of Global Weather 79](#_Toc422237351)

[Table 11‑3. Example of Regional Weather 79](#_Toc422237352)

[Table 11‑4. Example of Multiple Weather Regions 79](#_Toc422237353)

[Table 11‑5. Object/Entity Association Record 81](#_Toc422237354)

[Table 11‑7. Soil Attribute Enumeration 89](#_Toc422237355)

[Table 11‑8. Values in the Soil Attribute Map PDU Based on Example 90](#_Toc422237356)

[Table 11‑9. Soil Attribute Map PDU 93](#_Toc422237357)

[Table 12‑1. Jamming Technique Enumerations 96](#_Toc422237358)

[Table 12‑2. Designator PDU 97](#_Toc422237359)

[Table 12‑3. Laser Function Enumeration 98](#_Toc422237360)

[Table 12‑4. Designator Spot Type Enumeration 98](#_Toc422237361)

[Table 12‑5. Designator System Name Enumeration 98](#_Toc422237362)

# INTRODUCTION

Distributed Interactive Simulation (DIS) is a government/industry initiative to define an infrastructure for linking simulations of various types at multiple locations to create realistic, complex, virtual worlds for the simulation of highly interactive activities. DIS is widely used by a number of existing training systems. As newer versions are approved by the Institute of Electrical and Electronics Engineers (IEEE) and Simulation Interoperability Standards Organization (SISO) this VDIS specification will be updated to reflect the most recent current version (consolidating it with our existing extensions and modifications). The most current version to be approved is IEEE Std 1278.1-2012 on 19 December, 2012.

## Purpose

The Virtual Distributed Interactive Simulation (VDIS) specification provides the definition of the DIS extensions and modifications for SE Core CVEM. The VDIS team has analyzed a set of reference training systems from the U.S. Army Program Executive Office for Simulation, Training and Instrumentation (PEO STRI) virtual simulation training domain to determine differences in usage from the DIS standards. Recommended extensions and modifications to the standards are identified to define the DIS dialect to be used for SE Core interoperability. This document (and its associated enumerations reference documentation / files) defines the interoperability mechanisms required for products and sub-products at the interoperability layer to become SE Core compliant.

## Scope

The scope of VDIS is establishing the enhancements, extensions, and changes (possibly reverting) to IEEE Std 1278.1-2012 and the older IEEE Std 1278.2-1995. The IEEE documents will not be duplicated here and only the changes (with possible additional contextual fields or records) will be given. A revision at SISO of 1278.2 is currently ongoing and SE Core CVEM is involved in that effort in an attempt at ensuring compatibility with our requirements. VDIS also establishes a baseline of enumerations and works to make these compatible with the current enumerations baseline at SISO. The initial set was SISO-REF-010-2006, although the current SISO version is SISO-REF-010-2011.1 and a newer version SISO-REF-010-00v20-0 is nearly ready. The extensions to the DIS standard documents are provided in these Word documents, while the enumerations are provided in Excel files. Additionally, IEEE 1278.3 and 1278.4 will be deprecated in favor of Distributed Simulation Engineering and Execution Process (DSEEP) extensions.

Extensions and modifications made in VDIS will be submitted to the SISO standards committees for review and acceptance into the future DIS standards and enumerations documents. As the standards evolve, the current versions will be available through IEEE and SISO for use with SE Core products and sub-products.

# REFERENCE DOCUMENTS

Table ‑. Reference Documents

| Doc. Number | Title |
| --- | --- |
| SISO-REF-010-2006 | Simulation Interoperability Standards Organization (SISO) Enumeration and Bit Encoded Values for use with Protocols for Distributed Interactive Simulation Applications (EBV-DOC) |
| SISO-REF-010-2011 | Enumerations for Simulation Interoperability |
| SISO-REF-010-2011.1 | Enumerations for Simulation Interoperability |
| SISO-REF-010-00v20-0 | Enumerations for Simulation Interoperability (Draft CRs only) |
| IEEE 1278.2-1995 | IEEE Standard for Distributed Interactive Simulation – Communication Services and Profiles |
| IEEE 1278.3-1996 | IEEE Standard for Distributed Interactive Simulation – Exercise Management and Feedback |
| IEEE 1278.4-1997 | IEEE Standard for Distributed Interactive Simulation – Verification, Validation, and Accreditation |
| IEEE 1278.1a-1998 | IEEE Standard for Distributed Interactive Simulation – Application protocols (Supplement to IEEE Std 1278.1-1995) |
| IEEE 1278.1-2012 | IEEE Standard for Distributed Interactive Simulation – Application protocols |
| PCR 130I | Designator PDU revisions |
| IEEE 1730-2010 | Recommended Practice for Distributed Simulation Engineering and Execution Process |

# GENERAL REQUIREMENTS

This section defines the general requirements for VDIS.

## DIS Protocol Data Units (PDUs)

Protocol data units (PDUs) are data messages passed between the simulation applications according to the DIS protocols (with VDIS extensions and modifications). This section identifies the types of PDUs, families of PDUs, PDU header, general usage, issuance rules, receipt rules, heartbeats, and timeouts.

### Protocol Data Units

This section outlines the DIS PDUs that have been modified, extended, or added by VDIS. The enumerated value is listed in the parentheses to the right.

1. Entity Information/Interaction (1)
   1. Collision PDU (4) – Modified
2. Logistics (3) – Extensions to interaction rules and changes to timing constants
   1. Service Request PDU (5)
   2. Resupply Offer PDU (6)
   3. Resupply Received PDU (7)
   4. Resupply Cancel PDU (8)
   5. Repair Complete PDU (9)
   6. Repair Response PDU (10)
3. Simulation Management (5) – Extensions to interaction rules
   1. Create Entity PDU (11)
   2. Remove Entity PDU (12)
   3. Start/Resume PDU (13)
   4. Stop/Freeze PDU (14)
   5. Acknowledge PDU (15)
   6. Action Request PDU (16)
   7. Action Response PDU (17)
   8. Data Query PDU (18)
   9. Set Data PDU (19)
   10. Data PDU (20)
   11. Event Report PDU (21)
   12. Comment PDU (22)
4. Distributed Emission Regeneration (6)
   1. Electromagnetic Emission PDU (23)
   2. Designator PDU (24) - Modified
   3. IFF PDU (28)
5. Radio Communications (4)
   1. Transmitter PDU (25)
   2. Signal PDU (26)
   3. Receiver PDU (27)
6. Synthetic Environment (9)
   1. Environmental Process PDU (41) – Extended by records
   2. Point Object State PDU (43)
   3. Linear Object State PDU (44)
   4. Areal Object State PDU (45)
7. Experimental VDIS (130)
   1. Application Control PDU (200) – Added

### PDU Header

The VDIS PDU header record is not modified from the DIS PDU Header record except specific usages of values as noted in the following sections.

#### Protocol Version

The current version of the VDIS specification is based on IEEE 1278.1-2012. Although VDIS contains significant changes from the current DIS version, VDIS uses version 7 for the Protocol Version field.

#### Exercise Identifier

An Exercise Identifier with a value of NO\_EXERCISE shall be reserved for simulation management functions. An Exercise Identifier with a value of ALL\_EXERCISES shall be used to indicate that the PDU affects all exercises, which is typically a simulation management function as well.

The following are additional symbolic names:

|  |  |  |
| --- | --- | --- |
| Symbolic Name | Fixed or Variable | Fixed and Default Parameter Values |
| NO\_EXERCISE | Fixed | 0 |
| ALL\_EXERCISES | Fixed | FF (H) |

#### PDU Type

VDIS does not support the experimental CGF PDU Types 129-135. Additionally, VDIS has added the Application Control PDU (200).

#### Protocol Family

VDIS has added the Experimental VDIS (130) PDU Protocol Family. The Experimental – CGF (129) family was removed.

### PDU Usage, Issuance, and Receipt Rules

All PDUs that interest the after action review (AAR) simulation application shall be issued to allow receipt at all times, regardless of the multicast group segregation (so if the AAR is on a separate network, then the PDUs need to get propagated there). The definition of the interest of an AAR is currently outside the scope of this specification. Note that this issuance rule may also be applicable to other simulation applications and not just AARs, per exercise agreement.

Unless otherwise specified, the VDIS protocol follows the IEEE DIS protocols on issuance and receipt for the specific PDUs.

### Heartbeats and Timeouts

This section will be completed at a future release of this document. Some heartbeats and timeouts are handled individually, so refer to the PDU section for more details.

## Variable Parameter (VP) Records

The following table provides the list of VDIS-supported Variable Parameter (VP) Record Types, but note that the values beyond Attached Parts are subject to change based on an official enumerations revision:

Table ‑. Variable Parameters Record Type Enumeration

| Field Value | Parameter Type Designator |
| --- | --- |
| 0 | Articulated Part |
| 1 | Attached Part |
| 2 | Separation |
| 3 | Entity Type |
| 4 | Entity Association |
| 20 | Extended Platform Appearance |
| 21 | Extended Life Form Appearance |
| 22 | High Fidelity Lights |
| 23 | Chevron Marking (deleted) |
| 24 | High Fidelity Thermal Sensor |
| 25 | Entity Offset |
| 26 | Dead Reckoning |
| 27 | Army Task Organization (deleted) |
| 28 | Head Gazing / Weapon Aiming |
| 29 | Life Form Action Sequence |
| 30 | Legacy Extended Life Form Appearance |
| 31 | Extended Cultural Feature Appearance |
| 32 | Extended Supply Appearance |

Please refer to the individual sections describing the usage of the VP records for record-specific issuance and receipt rules. All Articulated Part and Attached Part VP records shall be transmitted prior to any other VP records. This will help to ensure compatibility with any legacy training systems capable of handling them without crashing. The next VP record in order should be an Extended Appearance VP record (if applicable) or other state-related VP records such as High Fidelity Thermal or High Fidelity Lights. Entity Association shall be transmitted prior to any Entity Offset or Dead Reckoning VP records that depend on the association.

## Coordinate Systems

VDIS implements the coordinate systems as described in IEEE Std 1278.1-2012, section 1.6.3. This version of DIS has incorporated the VDIS definition for object coordinate systems.

## Time Management

VDIS applications should use absolute time synchronized to universal coordinated time. VDIS recommends using Network Time Protocol (NTP) as the UTC time source as the exercise agreement. It is further recommended that NTPv3 or higher be supported. NTPv4 can usually maintain time within 10 milliseconds over a public internet and as accurate as 200 microseconds in a local area network under ideal conditions. Standard Positioning Service (SPS) is not yet required, but could be a later extension to support timing accuracy to 340 nanoseconds. SPS is the normal level of service offered by the Global Positioning System (GPS).

## Dead Reckoning Algorithm (DRA)

Except as noted below, VDIS adheres to the DIS standard regarding dead reckoning algorithms:

1. VDIS applications shall use a Dead Reckoning Algorithm of Static (1) when the entity is intended to be stationary. When the magnitudes of the linear velocity, linear acceleration, and angular velocity (or each component thereof) are effectively zero (very small) then the entity is stationary and the corresponding dead reckoning parameter should be set to 0. The actual value for “very small” is left for the individual applications to determine.
2. VDIS applications shall not use a Dead Reckoning Algorithm of Other (0).

# ENTITIES

## Entity Creation / Removal

Within VDIS all entities are created indirectly. Usage of the Create Entity-R or Create Entity PDU is not recommended. A simulation application shall create a new entity when an Entity State PDU is first received for an entity that was not previously created. The new entity identifier is contained in the Entity ID record of the Entity State PDU. Additionally, VDIS does not recommend usage of the Remove Entity-R or Remove Entity PDUs. Entity removal may occur based on timeout parameters, termination of the exercise, the Entity Appearance record State field (bit 23) set to Deactivated (1), or based on a pre-defined scenario. Entities identified as Live Entities shall never be subject to timeout, even if they no longer produce entity state information.

## Entity Identifier Record

The Site Number shall be unique and assigned during initialization. This may be addressed by the physical site or by type of training system but will most likely be a combination of the two. Please refer to the section on initialization for more information.

The Application Number shall be unique per application within a site which communicates within the VDIS network. VDIS does not currently support Serial Simulations (see section 4.2.3.3 of the DIS standard). The method of assignment of an Application Number to a simulation application is currently outside the scope of this specification.

## Force ID

All force ID enumerations shall be used and not reserved to support switching sides. For example, if Force X is currently Neutral 2, then Friendly 2 and Opposing 2 shall not be reserved to allow for Force X changing sides.

A side is defined as a Force ID Affiliation of exercise participants that share a common mission and have common goals. In VDIS a Force ID Affiliation contains the force ID of the side, the affiliation name, and the identification of the parent affiliation when one exists (Other means that it doesn’t exist). The sides are composed of forces and identify the major groupings of forces and their common relationships to other sides in a conflict. Note that sides are not necessarily symmetric. A side can contain other sides, forces, and/or units and entities that share the same view of other sides not in their side and force structure; the members of a side all interact with other sides with respect to this view, although sides within a side need not view each other as friendly. All members of the side should view members of their Force ID Affiliations or members of their affiliation’s parent or children affiliation chains as friendly.

In VDIS the Force ID Affiliation Record may be used to associate the Force IDs defined in an exercise with an affiliation. The Affiliation Name can be any text string up the maximum string length indicated in the record structure, but must also include a terminating null. This Force ID Affiliation Record, if available, is used to identify affiliations during scenario generation and can be used to express affiliations for Data Analysis Reporting. This record is not necessary if the force layout during scenario generation is correct for all participants or if the force affiliation is not pertinent to all participants. Please refer to Appendix A for the Force ID Affiliation Datum Record.

## Entity Type and Alternate Entity Type

The Entity Type will follow the VDIS enumeration tables. The current VDIS format includes source code compatible names (unique identifiers) as well as descriptive text (perhaps usable in code as text output for reporting) and an alt field (a reference to another entity type’s unique name, and not to be confused with the Alternate Entity Type field in the Entity State PDU). The expectation for an application receiving entities is to look up the Entity Type in their table. Upon failing to find the Entity Type, VDIS recommends one of the following three methods for resolving the failure in the receiving simulation, in order of preference:

1. Look up the alt field and, if non-zero, use that one. This process is continued recursively until a match is found (an entity is supported) or until the alt field contains a zero (or blank). If the alt field contains a zero then do not support that entity type.
2. Set the last non-zero field to zero and search again. This, in effect, rolls up the hierarchy. This should be done until the Category is reached, such that there are no valid entities of just Kind, Domain, and Country.
3. Do not support entity types that are not already included in the lookup database. This ensures 100% compatibility or a loss of functionality which is easily traced.

The Alternate Entity Type shall use the same enumerations in VDIS as the Entity Type. This field is used for Guise Mode.

## Entity Appearance

The Entity Appearance record is Kind and Domain specific and defined in the enumerations Excel file for appearance. Additionally, VDIS has proposed new variable parameter records to accommodate extended entity appearance for those issues not currently handled in the enumeration set. Please see section 3.2 for more information on the variable parameter records. These VP record appearance bitfields are also documented in the Excel file. The color-coding for the file is:

* Blank (white) – SISO-standardized
* Olive Green – SISO-standardized, but changes primarily encouraged by VDIS
* Brown – unused bits, separated by color for easier recognition
* Purple – Changes based on VDIS, but not yet in SISO
* Yellow – Changes to VDIS from previous version

### Land Platforms

To support lower fidelity simulations that do not want to provide separate entity information (via Entity Association VP record) for a motorcycle or bicycle (or similar Entity Type) rider, the Hatch appearance information may be used instead. Thus, when the Primary Hatch field (bits 9-11) indicates the hatch is open and person is visible (5), then a rider is present on the entity. For any other enumeration value, no rider will be presumed present.

The following is recommended for VDIS usage:

1. Civilian vehicles shall not be camouflaged. Therefore, set the Paint Scheme (bit 0) to Uniform Color (0) and the Camouflage Type (bits 17-18) to Desert (0). See the Extended Platform Appearance VP record in section 4.9.1 for more details.
2. Military vehicles shall not have a definitive camouflage pattern. Therefore, set the Paint Scheme (bit 0) to Uniform Color (0) and the Camouflage Type (bits 17-18) to Desert (0). See the Extended Platform Appearance VP record in section 4.9.1 for more details.

### Air Platforms

High fidelity landing gear capabilities shall be supported with articulated parts. Additional lighting and equipment capabilities are included in the Extended Appearance VP record (see section 4.9.1).

### Other Platforms and Munitions

No changes from the documented bitfield enumerations.

### Life Forms

VDIS recommends supporting the following Life Form States:

* Upright, standing still (1)
* Upright, walking (2)
* Kneeling (4)
* Prone (5)
* Crawling (6)
* Parachuting (8)
* Sitting (10)
* Crouching (12)
* Surrender (14)
* Detained (15)

Other Life Form States (Upright, running, Swimming, Jumping, Squatting, and Wading) may not be supported and, if transmitted by a simulation application, will be mapped to the default posture of Upright, standing still (1) or other enumerations as the receiving system decides.

### Environmental, Cultural Features, and Supplies

No changes from the documented bitfield enumerations.

### Radios, Expendables, and Sensor/Emitters

No changes from the documented bitfield enumerations.

## Entity Marking

The Entity Marking Field provides for the marking of the entity with a string that is interpreted for display. This consists of a character set definition and eleven characters within that set which may be interpreted (depending on the set). VDIS recommends supporting two character sets: ASCII and U.S. Army Marking. The Force ID field is used in conjunction with the Entity Marking when the U.S. Army character set is used for determining the actual enumeration set. The existing SISO-REF-010 Entity Marking enumerations for the updated U.S. Army Marking character set are out of date. SE Core CVEM (with CCTT concurrence) has submitted CR02565 to delete all of the original CCTT-based entity marking tables. This CR will be incorporated in SISO-REF-010-00v20-0. It would be too large an effort to update and especially maintain. Instead, VDIS is maintaining these enumerations until such a time as these can be investigated in the future, possibly with a combination task with Simulation To Mission Command Interoperability (SIMCI) efforts to improve entity initialization date. Please refer to section 8 for more information on Task Organization and the Entity Marking field.

## Capabilities

This field specifies the entity’s capabilities to perform certain tasks. The full list of bit-encoded Capabilities are shown in Table 4‑1. Bits 5-8 were added by VDIS and submitted as part of PCRs to the DIS PSG (but not yet as CRs to the Enumerations Working Group). SE Core CVEM also updated the text to be less protocol-specific and this is referenced in CR02647. This CR will be incorporated in SISO-REF-010-00v20-0.

Table ‑. Entity Capabilities

| Bit | Name | Meaning |
| --- | --- | --- |
| 0 | Ammunition Supply | Describes whether the entity is able to supply some type of ammunition in response to an appropriate service request. |
| 1 | Fuel Supply | Describes whether the entity is able to supply some type of fuel in response to an appropriate service request. |
| 2 | Recovery | Describes whether the entity is able to provide recovery (e.g., towing) services in response to an appropriate service request. |
| 3 | Repair | Describes whether the entity is able to supply certain repair services in response to an appropriate service request. |
| 4 | ADS-B | Describes whether the entity is equipped with Automatic Dependent Surveillance - Broadcast (ADS-B). |
| 5 | Sling Load Carrier | The Entity is able to carry a payload in a sling load. The extended appearance record (if available) will identify the current sling load status and entity association and/or entity offset records (if available) will provide additional sling load details (such as payload). |
| 6 | Sling Loadable | The Entity is able to be carried as a sling load payload. The extended appearance record (if available) will identify if it is currently sling loaded and entity association and/or entity offset records (if available) will provide additional sling load details (such as carrier). |
| 7 | IED Presence Indicator | The Entity is an IED or contains an IED. The extended appearance record (if available) will identify how well hidden the IED is on the Entity. An Attached Part (if applicable, for instance a jury-rigged munition does not apply here) will identify the IED explicitly. |
| 8 | Task Organizable | The Entity (normally a virtual manned module) can be task organized into an existing mixed mode unit (where mixed mode is intended to comprise a combination of computer-generated forces and virtual or even live forces). |
| 9-31 | N/A | Unused |

## Life Forms

VDIS recommends continued use of the legacy life form categories for humans until the life form schema revision is complete within the SISO EWG. High fidelity capabilities for head gazing and weapon aiming, action sequences, and extended appearance are presented in subsections below. Head gazing, weapon aiming, and action sequences aren’t required and aren’t yet supported by SE Core OneSAF Virtual. The (legacy) extended life form appearance record, however, is supported and required when the information contained therein is necessary. A revised schema for animals is also included here, but these enumerations will change in the revised SISO schema. This will occur after SISO-REF-010-00v20-0.

### Animal Life Forms

The assignment of fields for Animal life forms is defined in this section and as shown in the schema tables. New or additional animals should be added under the proper categories and subcategories or add new ones as needed. Where a zoology classification scheme is listed under Category, using a good web reference like Wikipedia will make it relatively easy to find the correct group to put a new animal enumeration under if it is not obvious from the groupings that already exist. An animal enumeration should only be listed once even though it could fit under several categories. As noted above, this schema is being reworked at SISO and will change in a future version.

1. **Kind.** Kind is always set to 3 (Life Form).
2. **Domain.** The domain of the animal is its primary habitat. Most birds will be included in the Air domain, despite the fact that they all will have to nest on land.
3. **Country.** Animals do not have a country associated with them. This field is set to zero (0).
4. **Category.** The category field represents the basic classification of an animal. When a zoology classification scheme is used, it does not mean that the Subcategory field will be a continuation of that scheme. The Subcategory may be a simplification or merely lists species or subspecies under the category. However, further groupings, if desired, may be achieved by designating specific ranges of Subcategory numbers for a group.
5. **Subcategory.** The Subcategory field is used to indicate a species or subspecies.
6. **Specific.** The Specific field is used to represent physical traits of an animal. This enumeration represents three attributes as follows:
   1. *Sex.* This attribute indicates the animal’s gender as either Male or Female. There is no option for unspecified. The Sex is defined in the most significant bit (7).
   2. *General Age Group.* This attribute identifies the age of the animal in a general grouping, such as newborn, young, and mature. The general age is defined as an enumeration across bits 4-6 of the Specific field, providing a possible 8 enumeration values. The value for 0 will be reserved for “Not specified.”
   3. *Color.* This attributes indicates the general color of the animal. A few general colors like black, brown, and white will be provided, as well as an option for “Based on species/breed (when only one color and pattern exists for an animal).” The color is defined as an enumeration across bits 0-3 of the Specific field, providing a possible 16 enumeration values. The value for 0 will be reserved for “Not specified.”
7. **Extra.** The Extra field is used to represent special animal data. This enumeration represents three attributes as follows:
   1. *Weight.* This attribute indicates the general weight of the animal. The weight is defined as an enumeration across bits 6-7 of the Extra field, providing a possible 4 enumeration values. The value for 0 will be reserved for “Not specified/Normal.”
   2. *Capability*. This attribute is used to indicate any special capabilities of an animal. The capability values depend on the type of animal. An example is a dog trained to detect explosives. If there are specific explosives that it is trained to detect, then metadata must be made available, such as a Variable Parameter record, to provide more specific information. The capability is defined as an enumeration across bits 3-5 of the Extra field, providing a possible 8 enumeration values. The value for 0 will be reserved for “Not Specified.”
   3. *Group*. This attribute indicates if the entity represents more than one animal. This is a special case that should only be used where there are large numbers of animals in a group that need to be seen visually or be detected by a sensor, there will be no physical interaction with other entities, and use of the Aggregate State PDU is not feasible. Metadata will need to be made available to define the exact quantity being represented, their distribution and other characteristics needed to support accurate visualization and sensor detection. Examples include a large or numerous flocks of birds, schools of fish, and herds of animals. Creating an aggregation using the Aggregate State PDU is the preferred way to indicate a small flock of birds, school of fish or herd of cattle. The group is defined as an enumeration across bits 0-2 of the Extra field, providing a possible 8 enumeration values. The value for 0 will be reserved for “Single Entity.”

### Dead Animals

Dead animals are needed to be represented to primarily support the use case for counter-IED training. Other use cases are possible, however, such as simply supporting clutter in various third-world environments. VDIS requires simply reusing the “live” Entity Type of the animal and setting the Health field of the Appearance record to Fatal Injury (3). While this might cause some extra work for the CLS operator to both create and then reset the state of the entity, it alleviates multiple Entity Types that are not used like that for anything else. A “dead animal” Entity Type isn’t recommended in the distribution simulation community.

### Head Gazing / Weapon Aiming

Head gazing and weapon aiming provide a difficult challenge for interoperability. The primary consideration is whether to simulate these functions as articulated parts, probably the most obvious and readily available means. There are two problems with using articulated parts, however. First, it costs 16 bytes per articulation parameter. At a basic level, the head will likely need to be animated with two degrees of freedom, similar to a tank’s turret and main gun. The neck can, of course, also tilt side to side but that movement is likely not necessary for current exercises. These two degrees of freedom therefore cost a total of 32 bytes. If you consider that a weapon could be aimed with the same number of degrees of freedom, then that would be another 32 bytes. The second problem is the latency of the head gazing / weapon aiming. It is critical in the exercise to realize if/when a potential enemy is tracking you with either their vision or their weapon. For the same reason the Designator PDU has a target, head gazing and weapon aiming should also have a target. Even with Dead Reckoning, it will be difficult to simulate this tracking over a network, but realize we have not even accounted for sufficient dead reckoning parameters of the tracking head / weapon within this VP record. Although not a problem in itself, a third situation arises depending on the implementation of action sequences. If articulated parts are not used for detailed action sequences, then using articulated parts for head gazing / weapon aiming would create a conflict.

Thus, using articulated parts for head gazing and weapon aiming is flawed and will likely not work well in an exercise. The solution is to provide a similar mechanism to the Designator PDU: identify the entity and/or object being tracked. For that purpose, we propose the following Head Gazing / Weapon Aiming VP record. This VP record shall be transmitted whenever the sending application deems it necessary. If not transmitted in the Entity State or Entity State Update PDU, then head gazing / weapon aiming parameters are assumed to revert back to their default states. This VP record need not be transmitted if the parameter would conflict with a Life Form Action Sequence VP record. If both are transmitted and a conflict is identified, then the Life Form Action Sequence VP record takes precedence.

The Head Gazing / Weapon Aiming VP record shall contain the following fields:

1. **Record Type.** This field shall identify the record as a Head Gazing / Weapon Aiming VP record. It shall be represented by an 8-bit enumeration.
2. **Head Gazing Type**. This field shall identify where the head is gazing (direction, station, or entity/object). It shall be represented by an 8-bit enumeration.
3. **Weapon Aiming Type.** This field shall identify where the weapon is aiming (direction, station, or entity/object). It shall be represented by an 8-bit enumeration.
4. **Out Station ID.** This field shall specify the station ID that the head is gazing out of (or through) and/or that the weapon is aiming out of (or through). For example, you would identify a particular window on a vehicle by station ID if the life form were aiming his weapon out that window. It shall be represented by a 16-bit unsigned integer.
5. **Tracked Station ID.** This field shall specify the station ID on the tracked entity/object. This provides a higher level of detail for tracking that just the entity/object, but less than global position. This field could be used to have a life form watching a particular doorway on a building or maybe the cockpit on an aircraft, for example. It shall be represented by a 16-bit unsigned integer.
6. **Tracked Entity/Object ID.** This field shall identify the entity using an Entity Identifier record or object using an Object Identifier record. Whether the target is an entity or object is denoted by the Head Gazing Type and/or Weapon Aiming Type. The two types shall not conflict.

Table ‑. Head Gazing / Weapon Aiming VP Record

| Field | Type |
| --- | --- |
| Record Type | 8-bit enumeration |
| Head Gazing Type | 8-bit enumeration |
| Weapon Aiming Type | 8-bit enumeration |
| Padding | 8-bits unused |
| Out Station ID | 16-bit unsigned integer |
| Tracked Station ID | 16-bit unsigned integer |
| Tracked Entity/Object ID | 48-bit record |
| Padding | 16-bits unused |

When the Head Gazing Type or Weapon Aiming Type indicates a tracked condition, the Tracked Entity/Object ID is tracked by the head gaze or the aiming of the weapon. Both Head Gazing Type and Weapon Aiming Type use the same enumeration values. It is the responsibility of the receiving simulation application to properly reposition the head / weapon to “point at” the correct entity / object. Enumeration values of Down, Up, Straight Ahead, Left, and Right provide general locations for when the other options are not used. A specific, global location (such as a terrain position) can also be specified, but a World Coordinates record or some other combination of records would need to be provided in a subsequent VP record or Attribute PDU (due to the size of a World Coordinates record).

The life form entity could be gazing out a particular station, as identified in the Out Station ID. This station ID will be on the station associated with the entity (via an Entity Association VP record) and must be the same for both the head gazing and weapon gazing (though only if both identify an out station). Similarly, the life form could be tracking a particular station on the tracked entity or object. This station is identified by the Tracked Station ID and must be the same for both head gazing and weapon aiming (if both are tracking).

#### Issuance Rules

Only one Head Gazing / Weapon Aiming VP record shall be transmitted per entity. This record does not currently support multiple weapons tracked on different targets. If the entity has multiple weapons held at the ready, then it is assumed that all such weapons would be targeting the same. The issuing application may discontinue issuing this record with the understanding that the receiving application will put the life form in a default state for the head gaze and weapon aiming, and that VDIS does not define the default state (it is beyond the scope of this specification).

#### Receipt Rules

There are no specific receipt rules for this VP record. The receiving application may use the data contained in the Head Gazing / Weapon Aiming VP record if it possesses the capability. This record is intended for high fidelity life forms, so it’s not expected to be useful for all training systems, especially legacy systems.

### Action Sequences

Another difficult challenge for simulating life forms is specifying action sequences. These are single, functional sequences such as getting into a vehicle, standing up, walking, or even vomiting. In a pure sense, these could be represented by many articulated parts, but the obvious problem with that is the sheer number of parts and articulated parameters for each. This may put serious pressure on the network, not to mention the size of the Entity State PDU if fully populated. A quick estimate of potential requirements for the number of articulated parameters: head (3 – rotate, bend, tilt), hands (6 – rotate, bend, and open), arms (8 – shoulder bend, lift, and rotate, elbow), torso (2 – twist and bend), legs (6 – leg lift and bend, knee), and feet (2 – lift). That equals 27 articulated parts and does not include if you also want each finger articulated separately or model the location of the eyes, tongue, toes, or any facial expressions. Modeling human facial expression is probably not feasible using articulated parts, so the rest of the human modeling should probably use another method, too. Also, note that any implementation for action sequences should not conflict with head gazing / weapon aiming.

The method that is probably best in terms of storage is to use an enumeration scheme. Enumerate all the actions or positions that life forms can perform and then identify which one(s) are being used by any existing entities. This type of system will suffer in the ideal case because it is not possible to enumerate all positions and actions for all life forms. However, we can somewhat easily enumerate those useful in our simulations and as training system developers work, more and more can be added later. By enumerating action sequences, it will also leave the level of fidelity up to the receiving application. A high fidelity simulation could become very detailed and realistic while a low fidelity simulation might simply cause a human to switch states (e.g., prone to standing).

In an effort to support these actions sequences, we propose the following Life Form Action Sequence VP record. This VP record is a type of animation control; though how the animation sequence is performed by the receiving application is not defined (it could be implemented in the IG or the host or not at all, depending on various factors including level of fidelity of the simulation). The loop mode, direction, and state all help identify the type of animation. Loop mode does not change through the life of the action sequence, though direction might. This VP record shall only be transmitted during an actual action sequence (just prior to start, during, and just after termination) or for as long as any high fidelity information (e.g., body position) is required.

The Life Form Action Sequence VP record shall contain the following fields:

1. **Record Type**. This field shall identify the record as a Life Form Action Sequence VP record. It shall be represented by an 8-bit enumeration.
2. **Action Control**.
   1. *Loop Mode*. This field identifies if the animation of the action sequence should repeat (continuous) or run and then terminate (one shot). It shall be represented by a 1-bit enumeration.
   2. *Animation Direction*. This field identifies the direction of the animation (forward or reverse). It shall be represented by a 1-bit enumeration.
   3. *Animation State*. This field commands the animation to start, stop, pause, or resume. It shall be represented by a 2-bit enumeration.
   4. *Action Direction*. This field identifies the direction of the action sequence (not the animation playback). It shall be represented by a 4-bit enumeration.
3. **Action Sequence Type**. This field identifies the action sequence. It shall be represented by a 16-bit enumeration.
4. **Body Position Type**. This field identifies the current body position of the life form. This field could be used in a lower fidelity model (which does not support complex or detailed action sequences) or perhaps to ensure synchronicity on the animation playback. It shall be represented by a 16-bit enumeration.
5. **Number of Frames**. This field identifies the total number of expected frames in the animation action sequence. It could be explicitly defined by exercise agreement, in the scenario file, or be used for a general status with respect to the total current Frame Count. It shall be represented by an 8-bit unsigned integer.
6. **Frame Count**. This field identifies the current frame in the animation action sequence. If the animation sequence is not identical across different simulation applications, then this field could still be used as a general status with respect to the total Number of Frames. It shall be represented by an 8-bit unsigned integer.
7. **Target Station ID**. This field identifies the station ID associated with the given Action Sequence Type (not Body Position Type). For example, this could define the door through which a life form exits a vehicle. This is used in conjunction with the Entity/Object ID. It shall be represented by a 16-bit unsigned integer.
8. **Entity/Object ID**. This field identifies the entity or object associated with the given Action Sequence Type. For example, this could identify the vehicle that a life form is attempting to enter. The Action Sequence Type will identify if the field refers to an Entity Identifier record or an Object Identifier record.

Table ‑. Life Form Action Sequence VP Record

| Field | Type |
| --- | --- |
| Record Type | 8-bit enumeration |
| Action Control | Loop Mode, 1-bit enumeration  (0=one shot, 1=continuous) |
| Animation Direction, 1-bit enumeration  (0=forward, 1=reverse) |
| Animation State, 2-bit enumeration  (0=Stop, 1=Pause, 2=Start, 3=Resume) |
| Action Direction, 4-bits enumeration |
| Action Sequence Type | 16-bit enumeration |
| Body Position Type | 16-bit enumeration |
| Number of Frames | 8-bit unsigned integer |
| Frame Count | 8-bit unsigned integer |
| Target Station ID | 16-bit unsigned integer |
| Entity/Object ID | 48-bit record |

Multiple action sequences could be operating at the same time for one entity (e.g., walking and chewing gum). The Body Position Type can be used to identify the current body position. If the Action Sequence is implemented by the receiving application, and a sequence is currently running, then the Body Position Type should be ignored. This field provides a means for low fidelity simulation to use this record for both sending and receiving (higher complexity positions perhaps without the higher complexity animations). The Number of Frames identifies the expected number of frames of the action sequence and most likely will be something designed in a common scenario file or external means (e.g., same vendor). Even if the total number of frames received is not identical, or if there is reason to believe that the originating entity animation is not similar, this field can still be used as a percentage adjustment when also given the Frame Count. Example Action Sequence Types include Standing Up, Falling Down, Crawling, Walking, etc.

#### Issuance Rules

Only one Life Form Action Sequence VP record shall be transmitted per entity. An issuing application is not required to transmit this record if it does not possess the capability. The issuing application may stop transmitting this record when no action sequence is taking place, but the final issuance of the record shall contain an Animation State of Stop (0).

#### Receipt Rules

There are no specific receipt rules for this VP record. The receiving application may use the data contained in the Life Form Action Sequence VP record if it possesses the capability. This record is intended for high fidelity life forms, so it’s not expected to be useful for all training systems, especially legacy systems.

### Legacy Extended Life Form Appearance VP Record

#### Purpose

Additional life form appearance information shall be communicated using the Legacy Extended Life Form Appearance VP record. This record is intended for high fidelity life forms, so it’s not expected to be useful for all training systems, nor is it required for all entities published by a simulation application. This record is intended for systems that do not take advantage of the new life form schema which uses a Category in the Entity Type field of 6 or higher or that still wish to support higher fidelity modeling of the legacy life form schema (Category 0-5).

NOTE—The bitfield record definitions for the Extended Equipment field are provided in an external Excel file.

#### Record Definition

The Extended Life Form Appearance VP record shall contain the following fields:

1. **Record Type**. This 8-bit enumeration shall identify the record as an Extended Life Form Appearance VP record.
2. **Clothing Scheme**. This 8-bit enumeration describes the clothing scheme for the life form. In a two-part clothing scheme, Primary is the top and Secondary is the bottom, where applicable.
3. **Primary Color**. This 8-bit enumeration specifies the color scheme if the Clothing Scheme is set to Solid Color (1). It will specify the color scheme of the first tone in a two tone clothing scheme. Otherwise, the Primary Color should be set to a closest match value in the case where the clothing scheme is not recognized. For example, this may contain a value for light brown for a sand camouflage clothing scheme.
4. **Secondary Color**. This 8-bit enumeration specifies the color scheme of the second tone in a two tone clothing scheme.
5. **Extended Equipment**. This 32-bit record defines the additional equipment that is visibly deployed by the life form.
6. **Status**. This 8-bit record defines special status indicators for the life form.
   1. *Invincible*. This 1-bit enumeration defines the invincibility characteristic of the life form.
   2. *Disguise*. This 3-bit enumeration defines how well the life form is disguised.
   3. *Present Domain*. This 4-bit enumeration defines the current domain for the life form, including a specific sub-domain.
7. **Attributes**. This 16-bit record defines the additional life form attributes.
   1. *General Race/Ethnicity*. This 4-bit enumeration defines the race of the life form, if applicable. A basic set of race/ethnicity choices are included. The intent of this attribute is to assist in picking an appropriate basic visual model for a human entity. Two options are provided (Homogenous and Indigenous) to indicate to use the Country field of the Entity Type. These categories are based primarily on the U.S. Office of Management and Budget (OMB, <http://www.whitehouse.gov/omb/>). Note, however, that race here does not directly determine country, so Hispanic as an ethnicity can be mixed in with the racial classifications. The objective is primarily to choose a visual model, not to identify a cultural background or speech pattern (though such a high fidelity simulation may require additional informational contained in a VP record). The East Asian Race/Ethnicity includes Chinese, Japanese, Koreans, Vietnamese, Philippines, Indonesians, Malayans, and Thais. The value for 0 for General Race/Ethnicity will be reserved for “Not specified.” General Race/Ethnicity will also include the following two values (in addition to other, specific races/ethnicities):
      1. Homogenous Country Code. The person represents the typical physical appearance of someone from a country that has a homogenous racial/ethnic appearance (e.g., Chinese, Japanese, etc.) The Country Code is not indicative of the race or ethnic appearance of someone from a country that has assimilated many different races and ethnic groups such as the United States.
      2. Indigenous Country Code. The person represents an indigenous person from a country that has only one indigenous group appearance (e.g., Aboriginals in Australia). The Country Code is not indicative of the race or ethnic appearance of someone from a country that has a number of indigenous with distinct physical differences. The United Nations defines Indigenous people as having a historical continuity with pre-invasion and pre-colonial societies that developed on their territories and consider themselves distinct from the rest of the societies that now prevail in those territories, or parts of them.
   2. *General Age Group*. This 4-bit enumeration defines the age of the person in a general grouping, such as newborn, teenager, and adult. The value for 0 will be reserved for “Not specified.”
   3. *Sex*. This 1-bit enumeration indicates the person’s gender as either Male or Female. There is no option for unspecified.

Table ‑. Legacy Extended Life Form Appearance VP Record

| Field | Type |
| --- | --- |
| Record Type | 8-bit enumeration |
| Clothing Scheme | 8-bit enumeration |
| Primary Color | 8-bit enumeration |
| Secondary Color | 8-bit enumeration |
| Extended Equipment | 32-bit record |
| Status | Invincible—1-bit enumeration |
| Disguise Status—3-bit enumeration |
| Present Domain—4-bit enumeration |
| Padding | 8 bits unused |
| Attributes | Padding—7 bits unused |
| General Race/Ethnicity—4-bit enumeration |
| General Age Group—4-bit enumeration |
| Sex—1-bit enumeration |
| Padding | 32 bits unused |

#### Issuance Rules

The following issuance rules apply:

1. Only one Legacy Extended Life Form Appearance VP record shall be transmitted per entity. An issuing application is not required to transmit this record if it does not possess the capability.
2. A Legacy Extended Life Form Appearance VP record shall not be used for Categories 6 and higher.
3. The issuing application may stop transmitting this record when no higher fidelity detail is provided by the record (but see below for specific field requirements that may override this). However, if the record is no longer issued, then the final issuance shall reflect the default state that matches the PDU without the record. For example, assume that a particular life form requires no further detail than the Present Domain change of being in the water (such as swimming). When the life form exits the water onto land, the final issuance of the Legacy Extended Life Form Appearance VP record would reflect a Present Domain of being on land (or unchanged). Some parameters in the Legacy Extended Life Form Appearance will require transmittance of the record throughout the existence of the entity, however. An example of this would be a specific color identified.

The following specific field requirements apply:

1. The Paint Scheme field of the Appearance record (bit 0) and the Camouflage Type of the Appearance record (bits 28-29) shall match the Legacy Extended Life Form Appearance VP record fields, if possible. For example, if the Legacy Extended Life Form Appearance VP record is used to indicate a camouflage clothing scheme of desert, winter, or forest camouflage, then the Appearance record shall also indicate that scheme. The Appearance record Camouflage Type shall be set to Other (3) when the Legacy Extended Life Form Appearance VP record indicates a camouflage clothing scheme that is not desert, winter, or forest.
2. The Paint Scheme field of the Appearance record (bit 0) shall be set to Uniform (0) for any Clothing Scheme other than camouflage.
3. If the Invincible field of the Status record is set to Invincible (1), then the life form does not sustain damage (by any means).

#### Receipt Rules

There are no specific receipt rules for this VP record. The receiving application may use the data contained in the Legacy Extended Life Form Appearance VP record if it possesses the capability.

#### VDIS Usage

This VP record will handle a more complete entity clothing scheme (solid colors in addition to a number of different, specialized paint schemes), decals, and some other attributes than the existing Appearance field. Example Clothing Schemes include Solid Color, Two Tone, and Camouflage.

## Platforms

### Extended Platform Appearance VP Record

#### Purpose

Additional platform appearance information shall be communicated using the Extended Platform Appearance VP record. This record is intended for high fidelity platforms, so it’s not expected to be useful for all training systems, especially legacy systems, nor is it required for all entities published by a simulation application. This VP record handles a more complete entity paint scheme (solid colors in addition to a number of different, specialized paint schemes) than the existing Appearance field. Example enumerations for Paint Scheme and Decal Scheme include Solid Color, Camouflage, name brands for mail delivery companies, other name brands, etc.).

NOTE—The bitfield record definitions for the Extended Lights, Thermal Indicators, and Extended Equipment fields are provided in an external Excel file.

#### Record Definition

The Extended Platform Appearance VP record shall contain the following fields:

1. **Record Type**. This 8-bit enumeration shall identify the record as an Extended Platform Appearance VP record.
2. **Paint Scheme**. This 8-bit enumeration describes the paint scheme for the platform.
3. **Decal Scheme**. This 16-bit enumeration describes the decals for the platform.
4. **Primary Condition/Material**. This 8-bit record defines the condition and material for the primary component platform. What constitutes the primary component depends on the platform and may be further specified by the paint scheme (such as truck and bed).
   1. *Rust.* This 2-bit enumeration indicates the condition of rust. See the table below.
   2. *Material.* This 2-bit enumeration indicates the material type. This will typically be metallic. Although there are a lot of different materials, the general objective here is to identify a common material that could denote a visual model. See the table below.
   3. *Exterior damage.* This 2-bit enumeration indicates the amount of exterior damage. This is not fire power kill or mobility kill damage and should not affect the function of the entity. This field can be used to identify a dented car, for instance, but probably not one damaged by a bomb. See the table below.
   4. *Clean.* This 2-bit enumeration indicates the cleanliness. See the table below.
5. **Primary Color**. This 8-bit enumeration specifies the color scheme if the Paint Scheme is set to Solid Color (1). It will specify the color scheme of the first tone in a two tone paint scheme. Otherwise, the Primary Color should be set to a closest match value in the case where the paint scheme is not recognized. For example, this may contain a value for light brown for a sand camouflage paint scheme. The Primary Color is also used to define the underlying color when a Paint Scheme indicates overlying patterns. For example, if the Paint Scheme identifies a Jingle (17) truck, then the Primary Color is used to identify the cab color.
6. **Secondary Condition/Material**. See Primary Condition/Material for record definition.
7. **Secondary Color**. This 8-bit enumeration specifies the color scheme of the second tone in a two tone paint scheme.
8. **Extended Lights**. This 32-bit record defines the additional lights on the platforms that were not handled in the Appearance field. This field is domain-specific.
9. **Thermal Indicators**. This 8-bit record describes the thermal components for the platform to aid in thermal sensors. This field is domain-specific. This is more than just an on/off flag for the power plant because even if the power plant is off, it could still give off a thermal signature.
10. **Status**. This 8-bit record defines special status indicators for the platform.
    1. *Invincible*. This 1-bit enumeration defines the invincibility characteristic of the platform.
    2. *Disguise*. This 3-bit enumeration defines how well the platform is disguised.
    3. *Present Domain*. This 4-bit enumeration defines the current domain for the platform, including a specific sub-domain.
11. **Extended Equipment**. This 16-bit record defines the additional equipment that is visibly deployed on the platform. This field is domain-specific.

Table ‑. Extended Platform Appearance VP Record

| Field | Type |
| --- | --- |
| Record Type | 8-bit enumeration |
| Paint Scheme | 8-bit enumeration |
| Decal Scheme | 16-bit enumeration |
| Primary Condition/Material | Rust, 2-bit enumeration |
| Material, 2-bit enumeration |
| Exterior Damage, 2-bit enumeration |
| Clean, 2-bit enumeration |
| Primary Color | 8-bit enumeration |
| Secondary Condition/Material | Rust, 2-bit enumeration |
| Material, 2-bit enumeration |
| Exterior Damage, 2-bit enumeration |
| Clean, 2-bit enumeration |
| Secondary Color | 8-bit enumeration |
| Extended Lights | 32-bit record (domain-specific) |
| Thermal Indicators | 8-bit record |
| Status | Invincible, 1-bit enumeration |
| Disguise, 3-bit enumeration |
| Present Domain, 4-bit enumeration |
| Extended Equipment | 16-bit record (domain-specific) |

#### Issuance Rules

The following issuance rules apply:

1. Only one Extended Platform Appearance VP record shall be transmitted per entity. An issuing application is not required to transmit this record if it does not possess the capability.
2. The issuing application may stop transmitting this record when no higher fidelity detail is provided by the record. However, if the record is no longer issued, then the final issuance should reflect the default state that matches the PDU without the record. For example, assume that a particular platform requires no further detail than the Present Domain change of being on the ground (such as a helicopter landing). When the platform takes off, the final issuance of the Extended Platform Appearance VP record would reflect a Present Domain of being in the air. Some parameters in the Extended Platform Appearance will require transmittance of the record throughout the existence of the entity, however. An example of this would be a specific color identified.

The following specific field requirements apply:

1. The Paint Scheme field of the Appearance record (bit 0) and the Camouflage Type of the Appearance record (bits 17-18) shall match the Extended Platform Appearance VP record fields, if possible. For example, if the Extended Platform Appearance VP record is used to indicate a camouflage paint scheme of desert, winter, or forest camouflage, then the Appearance record shall also indicate that scheme. The Appearance record Camouflage Type shall be set to Other (3) when then the Extended Platform Appearance VP record indicates a camouflage paint scheme that is not desert, winter, or forest.
2. The Paint Scheme field of the Appearance record (bit 0) shall be set to Uniform (0) for any Paint Scheme other than camouflage.
3. If the Invincible field of the Status record is set to Invincible (1), then the platform does not sustain damage (by any means).

#### Receipt Rules

There are no specific receipt rules for this VP record. The receiving application may use the data contained in the Extended Platform Appearance VP record if it possesses the capability.

#### VDIS Usage

Current VDIS usage of this record for color definition for single color civilian vehicles is as follows:

1. Set the Paint Scheme to Solid Color (1).
2. Set the Primary Color as appropriate, if known. If unknown, set the Primary Color to Not Specified (0).
3. Set the Secondary Color to Not Specified (0).
4. Set the Decal Scheme, Primary Condition/Material, and Secondary Condition/Material to zeroes.

Current VDIS usage of this record for color definition for multiple color civilian vehicles is as follows:

1. Set the Paint Scheme to Two Tone (Top, Bottom) (2), Two Tone (Body, Bed) (3), or Two Tone (Body, Trim) (4).
2. Set the Primary Color as appropriate, if known. If unknown, set the Primary Color to Not Specified (0).
3. Set the Secondary Color as appropriate, if known. If unknown, set the Secondary Color to Not Specified (0).
4. Set the Decal Scheme, Primary Condition/Material, and Secondary Condition/Material to zeroes.

Current VDIS usage of this record for color definition for military vehicles is as follows:

1. Set the Paint Scheme to Default (0).
2. If the scheme is known (i.e., tan or green), then set the Primary Color as indicated:
   1. Tan (98).
   2. Green (26).
3. If the scheme is not known (not defined), then set the Primary Color to Not Specified (0).
4. Set the Secondary Color to Not Specified (0).
5. Set the Decal Scheme, Primary Condition/Material, and Secondary Condition/Material to zeroes.

### Civilian Vehicles

VDIS has a laid out a new schema for civilian vehicles using Categories starting at 80 for each of the domains Land, Air, and Surface. With some modification, this schema was adopted in SISO-REF-010-00v20-0 and is reflected in the Entity Types and bitfield appearance enumerations.

#### Generic Vehicles / Roll-ups

Having a basic level of vehicle type discrimination at the Category level supports “roll-up” of more detailed types to this level for those simulations that have a minimal, but not non-existent, need to distinguish civilian vehicle types. Enumerating only to the category level allows distinguishing between a car, a bus, and a few other types of platforms like trucks and trailers, which are functionally and physically very distinct. This roll-up works within the VDIS paradigm as well, which supports roll-up to the Category level, but not to the Country level.

#### Additional Detail in VP Record

The Variable Parameter record will specify other details including color and possibly other visuals such as a door opening, spoilers, etc. Although some of these details may be available in the Extra field of the Entity Type, it’s our preference to utilize the Variable Parameter records and possibly the Capabilities field. For more information on appearance records, please see 4.9.1.5.

#### Multiple Choices

In some cases, there will be multiple methods to represent a similar entity. The best example of this would be a police car. You can choose to use the specific police car enumeration or you can define a generic make and model and then identify the police decal information and extended equipment in a variable parameter record. A high fidelity simulation may require multiple makes and models of police cars while a low fidelity simulation doesn’t, and therefore doesn’t require the usage of the extra 16 bytes for the variable parameter record.

#### Attached/Articulated Parts

A loader can use various attached (and articulated) parts. The same loader can use a bucket, auger, and other attachments. This obviously lends itself well to using the attached/articulated part mechanism for the attachments. These parts are not yet enumerated in the standard, but we have identified most of them. These attachments also apply to the various excavators and backhoe loader.

#### Extended Equipment

The list of extended equipment in the extended appearance variable parameter record should be revised to accommodate any known deficiencies from the civilian enumerations. For example, police lights should be part of the extended equipment and not an attached part (unless it is an undercover police car with a removable light).

The VDIS team has researched using the Vehicle Identification Number (VIN) system to define a particular civilian vehicle, but decided against it because the format is proprietary to the Society of Automotive Engineers (SAE) and is not suitable for simulation. The model year, in particular, is a poor choice because it does not have a definition prior to 1980 and the values start repeating in 2010. Most of the remaining information is either incomplete or given by both the SAE document and each of the manufacturers (i.e., potentially subject to change and very timeframe dependent).

## Extended Appearance for Other Entities

This section defines the extended appearance VP records for other kinds of entities besides platforms and life forms.

### Extended Cultural Feature Appearance VP Record

#### Purpose

Additional cultural feature appearance information shall be communicated using the Extended Cultural Feature Appearance VP record. This record is intended for high fidelity cultural features, so it’s not expected to be useful for all training systems, especially legacy systems, nor is it required for all entities published by a simulation application.

#### Record Definition

The Extended Cultural Feature Appearance VP record shall contain the following fields:

1. **Record Type**. This 8-bit enumeration shall identify the record as an Extended Cultural Feature Appearance VP record.
2. **Paint Scheme**. This 8-bit enumeration describes the paint scheme for the cultural feature. This paint scheme is the same as used for platforms.
3. **Primary Color**. Specifies the color scheme if the Paint Scheme is set to Solid Color (1). It will specify the color scheme of the first tone in a two tone paint scheme. Otherwise, the Primary Color should be set to a closest match value in the case where the paint scheme is not recognized. For example, this may contain a value for light brown for a sand camouflage paint scheme. It shall be represented by an 8-bit enumeration.
4. **Secondary Color**. Specifies the color scheme of the second tone in a two tone paint scheme. It shall be represented by an 8-bit enumeration.
5. **Status**. This 8-bit record defines special status indicators for the cultural feature.
   1. *Invincible*. This 1-bit enumeration defines the invincibility characteristic of the cultural feature. If set to Invincible (1), then the cultural feature does not sustain damage (by any means).
   2. *Disguise*. This 3-bit enumeration defines how well the cultural feature is disguised.
   3. *Present Domain*. This 4-bit enumeration defines the current domain for the cultural feature, including a specific sub-domain.
6. **Extended Equipment**. This 16-bit record defines the additional equipment that is visibly deployed on the cultural feature.

Table ‑. Extended Cultural Feature Appearance VP Record

| Field | Type |
| --- | --- |
| Record Type | 8-bit enumeration |
| Paint Scheme | 8-bit enumeration |
| Padding | 16-bits unused |
| Padding | 8-bits unused |
| Primary Color | 8-bit enumeration |
| Padding | 8-bits unused |
| Secondary Color | 8-bit enumeration |
| Padding | 32-bits unused |
| Padding | 8-bits unused |
| Status | Invincible, 1-bit enumeration |
| Disguise, 3-bit enumeration |
| Present Domain, 4-bit enumeration |
| Extended Equipment | 16-bit record |

#### Issuance rules

The following issuance rules apply:

1. Only one Extended Cultural Feature Appearance VP record shall be transmitted per entity. An issuing application is not required to transmit this record if it does not possess the capability.
2. The issuing application may stop transmitting this record when no higher fidelity detail is provided by the record. However, if the record is no longer issued, then the final issuance should reflect the default state that matches the PDU without the record. For example, assume that a particular cultural feature requires no further detail than the Present Domain change of being in the air (e.g., sling loaded). When the cultural feature is placed back on the ground, the final issuance of this record would reflect a Present Domain of being on the ground. Some parameters in this record may require transmittance of the record throughout the existence of the entity, however.

There are no specific field requirements for this record.

#### Receipt rules

There are no specific receipt rules for this VP record. The receiving application may use the data contained in the Extended Cultural Feature Appearance VP record if it possesses the capability.

### Extended Supply Appearance VP Record

#### Purpose

Additional supply appearance information shall be communicated using the Extended Supply Appearance VP record. This record is intended for high fidelity supplies, so it’s not expected to be useful for all training systems, especially legacy systems, nor is it required for all entities published by a simulation application.

#### Record definition

The Extended Supply Appearance VP record shall contain the following fields:

1. **Record Type**. This 8-bit enumeration shall identify the record as an Extended Supply Appearance VP record.
2. **Status**. This 8-bit record defines special status indicators for the supply.
   1. *Invincible*. This 1-bit enumeration defines the invincibility characteristic of the supply. If set to Invincible (1), then the supply does not sustain damage (by any means).
   2. *Disguise*. This 3-bit enumeration defines how well the supply is disguised.
   3. *Present Domain*. This 4-bit enumeration defines the current domain for the supply, including a specific sub-domain.
3. **Extended Equipment**. This 16-bit record defines the additional equipment that is visibly deployed on the supply.

Table ‑. Extended Supply Appearance VP Record

| Field | Type |
| --- | --- |
| Record Type | 8-bit enumeration |
| Padding | 8-bits unused |
| Padding | 16-bits unused |
| Padding | 32-bits unused |
| Padding | 32-bits unused |
| Padding | 8-bits unused |
| Status | Invincible, 1-bit enumeration |
| Disguise, 3-bit enumeration |
| Present Domain, 4-bit enumeration |
| Extended Equipment | 16-bit record |

#### Issuance rules

The following issuance rules apply:

1. Only one Extended Supply Appearance VP record shall be transmitted per entity. An issuing application is not required to transmit this record if it does not possess the capability.
2. The issuing application may stop transmitting this record when no higher fidelity detail is provided by the record. However, if the record is no longer issued, then the final issuance should reflect the default state that matches the PDU without the record. For example, assume that a particular supply requires no further detail than the Present Domain change of being in the air (e.g., sling loaded). When the supply is placed back on the ground, the final issuance of this record would reflect a Present Domain of being on the ground. Some parameters in this record may require transmittance of the record throughout the existence of the entity, however.

There are no specific field requirements for this record.

#### Receipt rules

There are no specific receipt rules for this VP record. The receiving application may use the data contained in the Extended Supply Appearance VP record if it possesses the capability.

## Present Domain

The following enumeration set defines the present domain of the entity. This is the domain in which the entity is currently (as of the last entity state) located. Examples of these are provided in the table for enhanced understanding. This enumeration is defined in the external Excel file, but replicated here for the additional explanation.

Table ‑. Present Domain Enumeration

| Value | Description | Example |
| --- | --- | --- |
| 0 | Unchanged | The Domain field of the Entity Type reflects the present domain that the entity is operating in |
| 1 | Land | An aircraft on land |
| 2 | Air | A paratrooper |
| 3 | Surface | A swimmer, seaplane on pontoons |
| 4 | Subsurface | A diver |
| 5 | Space | Astronaut |
| 6 | Subterranean Land | Human walking inside a cave or a vehicle going through a tunnel |
| 7 | Subterranean Air | Small UAV flying inside a cave or tunnel |
| 8 | Subterranean Surface | Human swimming on the surface of water inside a cave |
| 9 | Subterranean Subsurface | Human in scuba gear underwater below land in an aquifer |
| 10 | Subsurface Terrain | A vehicle or structure on the floor of an ocean |
| 11 | Subsurface Caves | Human in scuba gear underwater in a cave in the ocean |
| 12 | Interior Other Entity/Object | Human walking up stairs inside a building or driving a car |
| 13 | Exterior Other Entity/Object | Helicopter on the top of a building or ship |
| 14 | Administratively Inactive | Entity held off the battlespace, but not formally deactivated |
| 15 | Unused |  |

## High Fidelity Lights

This VP record was created to support a simulation requiring high fidelity lights. An example of this would be for an articulated spot light with special, possibly configurable, lighting parameters (as given in the record). This record, being a VP record, was obviously design for usage on entities and not environmental objects such as buildings (unless modeled as entities). To use this record for a light on an Object State, VDIS recommends attached an entity to the object via the Entity Association VP record and then also including the High Fidelity Lights VP record.

The High Fidelity Lights VP record shall contain the following fields:

1. **Record Type**. This field shall identify the record as a High Fidelity Lights VP record. It shall be represented by an 8-bit enumeration.
2. **Flags**. This field shall define various states of the light. It shall be represented by an 8-bit record of Boolean values. The defined flags are:
   1. *Bit 0* – Off (0) / On (1)
   2. *Bit 1* – Detached (0) / Attached (1)
   3. *Bit 2* – Direction: Omni directional (0) / Directional (1)
   4. *Bit 3* – Enable Feather
3. **Light Type**. This field shall define the type of light. The location of the light is implicit in the type unless a part ID is defined, where the light is obviously on the indicated attached/articulated part. This enumeration is TBD. It shall be represented by a 16-bit enumeration.
4. **Part ID**. This field shall define the articulated or attached part for the light. If the light is attached to a part, then the appropriate flag must be set to Attached. It shall be represented by an 8-bit unsigned integer. (Although articulated and attached part VP records require a 16-bit unsigned integer for the ID field, no part ID can ever be more than 255 due to the number of possible parts and the numbering scheme.)
5. **Color**. This field shall define the color of the light. It shall be represented by an 8-bit enumeration.
6. **Beam Half-Angle**. This field shall specify horizontal and vertical field of view (FOV) of the light, in degrees for directional lights (see the Direction flag). It shall be represented by an 8-bit unsigned integer.
7. **Range Attenuation**. This field shall specify attenuation range of the light, in meters. It shall be represented by a 16-bit unsigned integer.
8. **Intensity**. This field shall specify the scalar intensity of the light, with a value of 0 signifying no light (but perhaps not off, see the On/Off flag). It shall be represented by an 8-bit unsigned integer.
9. **Flash Rate**. This field shall specify the scalar interval flash rate of the light, with a value of 0 signifying that the light is steady, not flashing. It shall be presented by an 8-bit unsigned integer.
10. **Infrared Index**. This field shall specify the scalar infrared intensity of the light, with a value of 0 signifying that the light has no infrared properties (perhaps off). It shall be represented by an 8-bit unsigned integer.
11. **Thermal Index**. This field shall specify the scalar thermal index of the light, with a value of 0 signifying that the light has no thermal component (perhaps off). Note that a light that is on (see the Flags), but has no intensity (e.g., a dimmer switch turned very low) still may have a thermal index. It shall be represented by an 8-bit unsigned integer.
12. **Feather**. This field shall specify the scalar index for feathering the edge of the light, with 0 defining a hard edge and the maximum value defining a soft edge. Feathering must be enabled in the Flags field. It shall be represented by an 8-bit integer.

Table ‑. High Fidelity Lights VP Record – Values

| Field | Type |
| --- | --- |
| Record Type | 8-bit enumeration |
| Flags | 8-bit record of Booleans |
| Light Type | 16-bit enumeration |
| Part ID | 8-bit unsigned integer |
| Padding | 8-bits unused |
| Color | 8-bit enumeration |
| Beam Half-Angle | 8-bit unsigned integer |
| Range Attenuation | 16-bit unsigned integer |
| Intensity | 16-bit unsigned integer |
| Flash Rate | 8-bit unsigned integer |
| Infrared Index | 8-bit unsigned integer |
| Thermal Index | 8-bit unsigned integer |
| Feather | 8-bit unsigned integer |

#### Issuance Rules

Multiple High Fidelity Lights VP records may be transmitted per entity, with each one affecting a different light (or set of lights). An issuing application is not required to transmit this record if it does not possess the capability. The issuing application may stop transmitting this record when the light is turned off and no higher fidelity detail is provided or necessary by the record. If the record is no longer issued, the final issuance need not reflect the default state. For example, assume that this record is transmitted to provide a high fidelity thermal parameter for a search light. Even when that light is shut off, the thermal index may not immediately reach zero and thus this record shall continue to be transmitted until that higher fidelity detail is no longer available (the thermal index reaches a default state, presumably zero).

Once this record is transmitted, the Part ID shall not change. If it changes, it affects a different light and the old light may assume to be off.

#### Receipt Rules

There are no specific receipt rules for this VP record. The receiving application may use the data contained in the High Fidelity Lights VP record if it possesses the capability. This record is intended for high fidelity lights, so it’s not expected to be useful for all training systems, especially legacy systems.

## High Fidelity Thermal Sensing

This VP record was created to support a simulation requiring high fidelity thermal sensing.

The High Fidelity Thermal VP record shall contain the following fields:

1. **Record Type**. This field shall identify the record as a High Fidelity Thermal VP record. It shall be represented by an 8-bit enumeration.
2. **Solar Heating**. This field shall identify the scalar thermal index for the solar component relating to the entity. Usage of this field may depend on the receiver’s sensor model. It shall be represented by an 8-bit unsigned integer.
3. **Power Plant**. These fields shall identify the scalar thermal index for each of four power plants in or on the entity. They shall be represented by 8-bit unsigned integers. To help identify specific power plants on an entity, a clearer definition of each is provided below.
   1. *Power Plant 1*. This is the main engine, or engine #1, or port engine #1.
   2. *Power Plant 2*. This is the auxiliary engine, or engine #2, or starboard engine #1.
   3. *Power Plant 3*. This is engine #3 or port engine #2.
   4. *Power Plant 4*. This is engine #4 or starboard engine #2.
4. **Armament**. These fields shall identify the scalar thermal index for each of four armaments in or on the entity. They shall be represented by 8-bit unsigned integers. To help identify specific armament on an entity, a clearer definition of each is provided below.
   1. *Armament 1*. This is the main gun (left/port primary), such as the M256 120 mm smoothbore gun on an M1A2.
   2. *Armament 2*. This is the auxiliary gun #1 (right/starboard primary), such as the .50 cal. M2 machine gun in front of the commander's hatch on an M1A2.
   3. *Armament 3*. This is the auxiliary gun #2 (left/port secondary), such as the 7.62 mm M240 machine gun in front of the loaders hatch on an M1A2.
   4. *Armament 4*. This is the auxiliary gun #3 (right/starboard secondary), such as the 7.62 mm M240 machine gun on a coaxial mount on an M1A2.
5. **Drive train**. These fields shall identify the scalar thermal index for each of two drive trains in or on the entity. They shall be represented by 8-bit unsigned integers. To help identify specific drive trains on an entity, a clearer definition of each is provided below. Examples of these are the left and right tracks of a tracked vehicle such as the M1A1 or main and auxiliary (tail) rotors on a helicopter.
   1. *Drive Train 1*. This is the main drive train (left/top/port).
   2. *Drive Train 2*. This is the auxiliary drive train (right/bottom/starboard).

Table ‑. High Fidelity Thermal VP Record

| Field | Type |
| --- | --- |
| Record Type | 8-bit enumeration |
| Solar Heating | 8-bit unsigned integer |
| Power Plant 1 | 8-bit unsigned integer |
| Power Plant 2 | 8-bit unsigned integer |
| Power Plant 3 | 8-bit unsigned integer |
| Power Plant 4 | 8-bit unsigned integer |
| Armament 1 | 8-bit unsigned integer |
| Armament 2 | 8-bit unsigned integer |
| Armament 3 | 8-bit unsigned integer |
| Armament 4 | 8-bit unsigned integer |
| Drive Train 1 | 8-bit unsigned integer |
| Drive Train 2 | 8-bit unsigned integer |
| Padding | 32-bits unused |

#### Issuance Rules

Only one High Fidelity Thermal VP record shall be transmitted per entity. An issuing application is not required to transmit this record if it does not possess the capability. Once this record is transmitted, it may only be discontinued when the thermal data ends in a default state, which should be considered as the zero thermal value. The final issuance of this record should coincide with that default state.

#### Receipt Rules

There are no specific receipt rules for this VP record. The receiving application may use the data contained in the High Fidelity Thermal VP record if it possesses the capability.

## Mounting / Towing / Hoisting

VDIS has defined interaction rules for mounting, towing, and hoisting operations. These operations use Action Request and Action Response PDUs to establish the connection and the Variable Parameter records in section 4.14.3 to provide the necessary additional information for the respective Entity State PDUs.

### Mounting

#### Introduction

The requirements for mounting and dismounting operations are defined in this section.

For convenience, this section will refer to the entity carrying the mounted entities/objects as the “carrier.” The entity or object being carried will be referred to as the mounted entity or object. General requirements for all mount/dismount operations are defined in 4.14.1.2. Specific requirements to capability discernment, mounting, and dismounting are defined in following sections.

#### General Requirements

The following general requirements shall apply to all mounting operations:

1. The carrier shall not initiate the mount/dismount operation.
2. The response to any mount or dismount Action Request PDU shall be performed with the Action Response PDU.
3. The Action Response PDU shall be issued within 10 seconds of receipt of the Action Request PDU.
4. The application sending the request shall timeout the request if a response is not provided in time.
5. The sending application shall reset this timer whenever an Action Response PDU with a Request Status of Pending (1) or Executing (2) is received. Note that Pending (1) may not be supported (see below).
6. If the time to process an Action Request PDU is expected to take longer than 10 seconds, then the receiving application shall issue an interim Action Response PDU with a Request Status of Executing (1).
7. No datum records shall accompany an Action Response PDU with a Request Status of Pending (1) or Executing (2).

#### Capability Discernment

The following requirements shall apply to all mounting operations:

1. Mount capability may be determined by the carrier’s current associations and comparing them with the number of a priori known capabilities.
2. Current ability to mount should be discerned by looking at the carrier’s Appearance field in the Entity State PDU. For example, an Armored Personnel Carrier (APC) should have the Ramp (bit 25) set to Down (1) before allowing life forms to mount. Which fields are required are entity specific and may also use the extended appearance data available in variable parameter records.

#### Mounting

The following requirements shall apply to all mount requests and following successful mounting:

1. The mount request shall be performed by issuing an Action Request PDU with an Action ID of Mount (18) and using the Number to Mount/Dismount (16210), Carrier Mount ID (16220), and Mount/Dismount Data (16215) datum records.
2. If more than 32 entities or objects need to mount at one time, then multiple mount operations shall be used.
3. The Request Status field in the Action Response PDU should be set to Executing (2), Complete (4), or Request Rejected (5).
4. Upon a successful mounting, the following additional rules apply:
   1. If the mounted entity is a life form then it shall set the Mounted/Hoisted flag in the Appearance field of its Entity State PDU to Mounted/Hoisted (1)
   2. If the simulation application for the mounted entity (not object) supports the Extended Platform Appearance VP record for platforms or (Legacy) Extended Life Form Appearance VP record for life forms, then it shall modify the Status record, Present Domain field to Interior Other Entity/Object (12) or Exterior Other Entity/Object (13).
   3. The carrier shall start transmitting an Entity Association VP record in its Entity State PDU. This record shall set the following fields accordingly:
      1. The Association Status shall be Physical Association Carrier (4).
      2. The Association Type shall be Mounted, with the exact enumeration depending on detailed operations, as follows:
         1. Mounted Attached (5) – The mounted entity is physically attached. For example, using a seatbelt in a car. Cargo is usually physically connected. This option also assumes that the mounted entity or object is supported. For this Association Type, the Connection Type shall be one of the options that specify the physical connection type, such as Rope (3) or Chain (4). The option for Attached Directly to Surface (1) may be used if the cargo (e.g.) is bolted down and not simply tied down.
         2. Mounted Unattached and Unsupported (6) – The mounted entity is neither physically attached nor directly supported by the carrier. For example, jumping up in the air on the deck of a ship. This option is rare and typically requires a high fidelity model.
         3. Mounted Unattached and Supported (7) – The mounted entity is not physically attached, but is directly supported by the carrier. For example, sitting in a seat but not belted in or standing on the bridge of a ship.
      3. The Connection Type shall be set appropriately for the Association Type. Examples (recommendations) are provided below:
         1. Mounted Attached (5) – Rope (3)
         2. Mounted Unattached and Unsupported (6) – Not specified (0)
         3. Mounted Unattached and Supported (7) – In Contact With (11)
      4. The Associated Object-2 Identifier shall specify the mounted entity or object.
   4. The mounted entity (not object) shall start transmitting an Entity Association VP record in its Entity State PDU. This record shall set the following fields accordingly:
      1. The Association Status shall be Physical Association Target (1).
      2. The Association Type shall be the same as specified by the carrier.
      3. The Connection Type shall be the same as specified by the carrier.
      4. The Associated Object-2 Identifier shall specify the carrier.
   5. The carrier shall transmit one or more Entity Offset VP records following the Entity Association VP record. One of these records shall denote the mounted entity or object position. Another may be issued to denote orientation of the mounted entity or object.
   6. Receiving applications shall use the position (and possibly orientation) as specified or modified by the carrier. Typically, this will mean that the position in the mounted entity Entity State PDU or in the mounted object Object State PDU is not used.
   7. The carrier may transmit one or more Dead Reckoning VP records to provide high fidelity dead reckoning parameters for the mounted entity or object position. If these Dead Reckoning VP records are not provided, then receiving simulation applications shall use the dead reckoning parameters of the carrier to update the position of the mounted entity or object, or dead reckon the carrier before applying the position offset.
5. The following additional rules apply with respect to damage:
   1. The mounted entity or object shall continue to model battle and collision damage for itself, as it normally would.
   2. The carrier shall model damage for any connecting apparatus (e.g., rope), if that sort of damage is modeled.
6. The following additional rules apply with respect to collisions, if collision detection is supported by the simulation application simulating the carrier:
   1. The carrier shall perform collision detection for the connecting apparatus and mounted entity or object.
   2. The carrier shall output the Collision PDU when a collision is detected with itself, another entity, another object, or the terrain.
   3. When the carrier outputs the Collision PDU for a collision with the mounted entity or object, then it shall set the Collision Origin field to Collision with mounted entity (3).
   4. If the collision occurs with the carrier itself, then no second Collision PDU shall be transmitted.

#### Dismounting

The following requirements shall apply to all dismount requests and following successful dismounting:

1. The dismount request shall be performed by issuing an Action Request PDU with an Action ID of Dismount (19) and using the Number to Mount/Dismount (16210), Carrier Mount ID (16220), and Mount/Dismount Data (16215) datum records.
2. The Request Status field in the Action Response PDU should be set to Executing (2), Complete (4), or Request Rejected (5).
3. Upon a successful dismounting, the following additional rules apply:
   1. If the mounted entity is a life form then it shall set the Mounted flag in the Appearance field of its Entity State PDU to Not mounted (0)
   2. If the simulation application for the mounted entity (not object) supports the Extended Platform Appearance VP record for platforms or (Legacy) Extended Life Form Appearance VP record for life forms, then it shall modify the Status record, Present Domain field to a suitable value, such as Unchanged (0). Another example would be a paratrooper exiting an aircraft where the Present Domain becomes Air (2).
   3. The carrier shall send out a revised Entity State PDU as follows:
      1. It shall include a final Entity Association VP record with an Association Status set to Association Broken (3).
      2. It shall no longer include Entity Offset VP or Dead Reckoning VP records (for this mounting operation).
   4. The carrier shall no longer perform collision detection for the mounted entity or object.
   5. The mounted entity (not object) shall include a final Entity Association VP record with an Association Status set to Association Broken (3).

### Towing

#### Introduction

The requirements for towing operations are defined in this section.

For convenience, this section will refer to the entity performing the towing as the “carrier.” The entity being towed will be referred to as the towed entity. General requirements for all towing operations are defined in 4.14.2.2. Specific requirements to hitching and unhitching are defined in following sections.

#### General Requirements

The following general requirements shall apply to all towing operations:

1. Either the carrier or the towed entity may initiate the towing operation.
2. The response to any hitch or unhitch Action Request PDU shall be performed with the Action Response PDU.
3. The Action Response PDU shall be issued within 10 seconds of receipt of the Action Request PDU.
4. The application sending the request shall timeout the request if a response is not provided in time.
5. The sending application shall reset this timer whenever an Action Response PDU with a Request Status of Pending (1) or Executing (2) is received. Note that Pending (1) may not be supported (see below).
6. If the time to process an Action Request PDU is expected to take longer than 10 seconds, then the receiving application shall issue an interim Action Response PDU with a Request Status of Executing (1).
7. No datum records shall accompany an Action Response PDU with a Request Status of Pending (1) or Executing (2).

#### Capability Discernment

The following requirements shall apply to all towing operations:

1. An entity capable of towing shall set its Recovery bit in the Entity Capabilities field (bit 2) to True (1).

#### Hitching

The following requirements shall apply to all towing requests and following successful hitching:

1. The towing request initiated by the towed entity shall be performed by issuing an Action Request PDU with an Action ID of Initiate Hitch Lead (15).
2. The towing request initiated by the carrier shall be performed by issuing an Action Request PDU with an Action ID of Initiate Hitch Follow (16).
3. No datum records should be sent in the Action Request PDU or Action Response PDU for towing operations.
4. The Request Status field in the Action Response PDU should be set to Executing (2), Complete (4), or Request Rejected (5).
5. Upon a successful hitching, the following additional rules apply:
   1. The carrier shall start transmitting an Entity Association VP record in its Entity State PDU. This record shall set the following fields accordingly:
      1. The Association Status shall be Physical Association Carrier (4).
      2. The Association Type shall be Towed, with the exact enumeration depending on detailed operations, as follows:
         1. Towed on Land (2) – The entity is towed on land. This is the most common type of towing that most people are familiar with (e.g., a tow truck).
         2. Towed on Water Surface (3) – The entity is towed on a body of water, floating.
         3. Towed Underwater (4) – The entity is towed underwater.
         4. All Towed In Air enumerations shall be restricted to Sling Load operations.
      3. The Connection Type shall be set appropriately.
      4. The Associated Object-2 Identifier shall specify the towed entity.
   2. The towed entity shall start transmitting an Entity Association VP record in its Entity State PDU. This record shall set the following fields accordingly:
      1. The Association Status shall be Physical Association Target (1).
      2. The Association Type shall be the same as specified by the carrier.
      3. The Connection Type shall be the same as specified by the carrier.
      4. The Associated Object-2 Identifier shall specify the carrier.
   3. The carrier shall transmit one or more Entity Offset VP records following the Entity Association VP record. One of these records shall denote the towed entity position. Another may be issued to denote orientation of the towed entity.
   4. Receiving applications shall use the position (and possibly orientation) as specified or modified by the carrier. Typically, this will mean that the position in the towed entity Entity State PDU is not used.
   5. The carrier may transmit one or more Dead Reckoning VP records to provide high fidelity dead reckoning parameters for the towed entity position. If these Dead Reckoning VP records are not provided, then receiving simulation applications shall use the dead reckoning parameters of the carrier to update the position of the towed entity, or dead reckon the carrier before applying the position offset.
6. The following additional rules apply with respect to damage:
   1. The towed entity shall continue to model battle and collision damage for itself, as it normally would.
   2. The carrier shall model damage for any connecting apparatus (e.g., rope), if that sort of damage is modeled.
7. The following additional rules apply with respect to collisions, if collision detection is supported by the simulation application simulating the carrier:
   1. The carrier shall perform collision detection for the connecting apparatus and towed entity.
   2. The carrier shall output the Collision PDU when a collision is detected with itself, another entity, another object, or the terrain.
   3. When the carrier outputs the Collision PDU for a collision with the towed entity, then it shall set the Collision Origin field to Collision with towed entity (5).
   4. If the collision occurs with the carrier itself, then no second Collision PDU shall be transmitted.

#### Unhitching

The following requirements shall apply to all unhitch requests and following successful unhitching:

1. The unhitch request initiated by either the carrier or the towed entity shall be performed by issuing an Action Request PDU with an Action ID of Unhitch (17).
2. No datum records should be sent in the Action Request PDU or Action Response PDU for towing operations.
3. The Request Status field in the Action Response PDU should be set to Executing (2), Complete (4), or Request Rejected (5).
4. Upon a successful unhitching, the following additional rules apply:
   1. The carrier shall send out a revised Entity State PDU as follows:
      1. It shall include a final Entity Association VP record with an Association Status set to Association Broken (3).
      2. It shall no longer include Entity Offset VP or Dead Reckoning VP records (for this towing operation).
   2. The carrier shall no longer perform collision detection for the towed entity.
   3. The towed entity shall include a final Entity Association VP record with an Association Status set to Association Broken (3).

### Hoisting

#### Introduction

The requirements for hoisting operations are defined in this section. Hoisting operations are those such as a helicopter hoisting dismounted infantry, combat medics, rescue personnel, or patients up into the cabin or to ride alongside the helicopter.

For convenience, this section will refer to the entity performing the hoisting as the “carrier.” The entity being hoisted will be referred to as the hoisted entity. General requirements for all hoisting operations are defined in 4.14.3.2.

When station identifiers become identified in VDIS (or DIS), an ID needs to be defined for the hoist, per helicopter. Consider a future capability to allow for a line that is able to extend (the hoist line). This is different from sling load because the line length does not change in sling load operations. A possible approach would be an articulated part, which would require definition of an articulated part enumeration. Another possible approach would be an Entity Association VP record identifying the ownship in the Associated Object-2 Identifier field and Entity Offset VP records identifying the endpoints of the line.

#### General Requirements

The following general requirements shall apply to all hoisting operations:

1. The carrier shall not initiate the hoist operation.
2. The response to any mount or dismount Action Request PDU shall be performed with the Action Response PDU.
3. The Action Response PDU shall be issued within 10 seconds of receipt of the Action Request PDU.
4. The application sending the request shall timeout the request if a response is not provided in time.
5. The sending application shall reset this timer whenever an Action Response PDU with a Request Status of Pending (1) or Executing (2) is received.
6. An Action Response PDU with a status of Pending (1) indicates the carrier is performing actions in preparation for the hoist activity. No further action, other than resetting the timer, is expected of the sending application at this point.
7. An Action Response PDU with a status of Executing (2) indicates the carrier is currently hoisting the entity. The sending application shall add an Entity Association VP record as specified below. The carrier shall add Entity Association and Entity Offset VP records for the hoisted entity.
8. If the time to process an Action Request PDU is expected to take longer than 10 seconds, then the receiving application shall issue an interim Action Response PDU with the same Request Status sent last.
9. An Action Response PDU may contain additional datum records intended to enhance the hoist operation.
   1. A suggested posture for the hoisted entity can be included as a Fixed Datum Record – DI Stance ID (16206). When present and supported by the hoisted entity, the hoisted entity shall change their posture to the requested state.
   2. If the carrier publishes the hoist device on the network, a Variable Datum Record – Carrier Mount ID (16220) – can be included indicating the device id the carrier intends to use for the hoist operation. When present and supported by the hoisted entity, the hoisted entity shall utilize the device id when publishing the Entity Association VP record indicating it is hoisted.  
        
      NOTE—The carrier shall examine the Entity Association VP records of the hoisted entity. If one of the records indicates the requested hoist device, then the hoist device shall publish Entity Association and Entity Offset VP records for the hoisted entity. Otherwise, the carrier shall publish these records for the hoisted entity.
10. Hoisting operations shall use the mounting rules regarding Entity Association VP records except as follows:
    1. When the hoisted entity is not yet loaded into the aircraft (not using straps):
       1. The (Physical) Association Type shall be Hoisted (15).
       2. The (Physical) Connection Type shall be Cable Line (7).
    2. When the hoisted entity is loaded into the aircraft:
       1. The (Physical) Association Type shall be Mounted Unattached and Supported (7).
       2. The (Physical) Connection Type shall be In Contact With (11).
11. Multiple life forms may be on the hoist, up to the normal limit for mounting operations and further restricted by the individual entities involved (via the behaviors; which is outside the scope of this specification).
12. If two or more life forms are hoisted on the same entity, then they shall have the same container that is able to support multiple life forms (e.g., jungle penetrator can support two).
13. The following rules are regarding Hoist Status:
    1. The Hoist Status field of the Extended Equipment record for Air Platforms (within the Extended Platform Appearance VP record) shall have the following enumeration meanings:
       1. Hoist Not Present – The hoist apparatus is not present or not defined (support for legacy)
       2. Hoist Unavailable – Hoist cannot service a hoist request at this time. (e.g., the hoist is in transit and not available to service mount requests, or the hoist is full)
       3. Hoist Available – Hoist can accept hoist requests at this time
    2. The current Hoist Status and Weight on Wheels setting of the helicopter shall determine whether an entity is mounting directly to the helicopter or to the hoist apparatus.
       1. If the helicopter has Weight on Wheels, then all mount operations shall target the helicopter itself regardless of the Hoist Status
       2. If the hoist is present (whether available or not), then all mount operations shall target the hoist.
       3. If the hoist is not present, then all mount operations shall target the helicopter itself.
       4. If the hoist is present but unavailable, then all mount operations shall fail.

#### Capability Discernment

The following requirements shall apply to all hoisting operations:

1. Hoist capability may be determined by the carrier’s current associations and comparing them with the number of a priori known capabilities.
2. Current ability to hoist should be discerned by looking at the carrier’s Hoist Status field of the Extended Equipment record for Air Platforms (within the Extended Platform Appearance VP record).

#### Hoisting

Hoisting operations shall follow the requirements identified in the previous section with the following exceptions / additions:

1. The Request Status field in the Action Response PDU should be set to Pending (1), Executing (2), Complete (4), or Request Rejected (5).
2. When the Request Status field of an Action Response PDU indicates Executing (2), the hoisted entity shall transmit an Entity Association VP record in its Entity State PDU. This record shall set the following fields accordingly:
   1. The Association Status shall be Physical Association Target (1).
   2. The Association Type shall be Hoisted (15).
   3. The Connection Type shall be Cable Line (7)
   4. The Associated Object-2 Identifier shall specify either the carrier or the carrier’s hoist device as indicated within a Carrier Mount ID (16220) Variable Datum record within the Action Response PDU.
3. If the hoisted entity is a life form then it shall set the Mounted/Hoisted flag in the Appearance field of its Entity State PDU to Mounted/Hoisted (1) while being hoisted
4. If the simulation application for the hoisted entity supports the Extended Platform Appearance VP record for platforms or (Legacy) Extended Life Form Appearance VP record for life forms, then it shall modify the Status record, Present Domain field to Exterior Other Entity/Object (13) while being hoisted.
5. When setting the Request Status field in the Action Response PDU to Executing (2), the carrier shall:
   1. Transmit Entity Association and Entity Offset VP records within either its Entity State PDU or the Entity State PDU of its hoist device. The carrier shall publish the records within the hoist device’s Entity State PDU if, and only if, the hoisted entity indicates it is associated with the hoist device. Otherwise, the records shall be published within the carrier’s Entity State PDU. The Entity Association record shall set the following fields accordingly:
      1. The Association Status shall be Physical Association Carrier (4)
      2. The Association Type shall be Hoisted (15)
      3. The Connection Type shall be Cable Line (7)
      4. The Associated Object-2 Identifier shall specify the hoisted entity

In Figure 4‑1 below, the arrowed lines indicate Entity Association VP records while the hoist is in the Executing state.

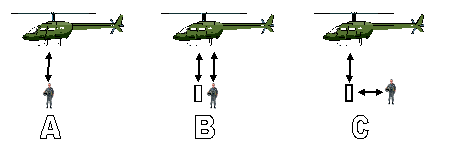


Figure ‑. VDIS Hoisting

Leftmost diagram (A) depicts a simplified hoist operation with no hoist device. The carrier has an Entity Association VP record identifying the hoist entity, and the hoist entity has an Entity Association VP record identifying the carrier.

Center diagram (B) depicts a hoist operation with a hoist device. In this case, the hoisted entity did not support the reception of a Carrier Mount ID record within the Action Response PDU. As such, the carrier Entity State PDU has two Entity Association VP records, one identifying the hoist device, and the other identifying the hoist entity. The carrier also publishes unique Entity Offset VP records following each Entity Association VP record. Both the hoist device and the hoist entity publish an Entity Association VP record indicating the carrier.

Rightmost diagram (C) depicts a hoist operation with a hoist device. In this case, the hoisted entity supports the reception of a Carrier Mount ID record identifying the hoist device. For this case, the carrier publishes Entity Association and Entity Offset VP records for the hoist device within the carrier Entity State PDU. The hoist device Entity State PDU contains an Entity Association VP record indicating it is being hoisted by the carrier. The hoist device also publishes Entity Association and Entity Offset VP records indicating it is hoisting the hoisted entity. The hoisted entity publishes an Entity Association VP record indicating it is being hoisted by the hoist device.

## Associated Entities

### Entity Association VP Record

To support mounting, towing, sling loads, and any operations requiring physical connections, VDIS adopts the Entity Association VP record as detailed in DIS. An issuing application may produce multiple Entity Association VP records for one entity (it may be a many-to-many relationship). VDIS currently only recommends this record for physical associations and not functional associations.

VDIS extends this record in one important way. The Change Indicator field also applies to the two (potential) dependent records: the Entity Offset VP record (see section 4.15.2) and the Dead Reckoning VP record (see section 4.15.3). If either of those records exist (including multiples) that apply to a particular Entity Association VP record, then the Change Indicator field for that Entity Association VP record will also be updated when any of those dependent records change.

When looking at these enumerations, the “Target” is the entity/object being acted upon (e.g., the one being towed, carried or is mounting) and the “Carrier” is the entity/object doing the towing, carrying, or acting. In cases of mutual physical association, both may be the target or carrier.

The following illustrates an example of a CH-47D with a rotor articulated part (not required on every use case, but presented here for context), extended appearance, and two sling load attachments. The Entity Offset and Dead Reckoning VP records are indented for clarity (refer to the sections describing their issuance rules).

Table ‑. Entity Association Example (RWA)

| Field | Notes |
| --- | --- |
| Entity State PDU | RWA (CH-47D) |
| Articulated Part | Rotor – Azimuth |
| Articulated Part | Rotor – Azimuth Rate |
| Extended Appearance | Air Platform |
| Entity Association | Center Hook Sling Load (low fidelity) |
| Entity Offset | Position |
| Entity Association | Aft Hook Sling Load (high fidelity) |
| Entity Offset | Position |
| Dead Reckoning | Linear Velocity |
| Dead Reckoning | Linear Acceleration |
| Entity Offset | Orientation |
| Dead Reckoning | Angular Velocity |

### Entity Offset VP record

This scheme for entity association does not account for an offset or orientation of the entity, however. For example, one vehicle can tow another using a solid connection or a line. If solid, then the offset will be static, but if it is a line (cable of some sort), then the offset will be dynamic and it should not be the receiving entities’ responsibility to perform the calculations of the position. Also, the reason that the Entity Association VP record is needed in the first place negates using the position of the towed entity (dead reckoning hiccups, bungee-like positioning and orientation).

What is really needed is an offset and orientation. Therefore, we propose an Entity Offset VP record as given below. This offset is not only used for towing and mounting. It may also be used for sling loads or to provide other types of offsets (see the enumeration for Offset Type). For these records (regardless of use case), we refer to the entity that is carrying or towing the other entity as the *carrier*. The entity that is being towed or carried is the *mounted entity*. Both the carrier and the mounted entity can produce this record.

#### Purpose

Identifying an offset or orientation of an entity shall be communicated with the Entity Offset VP record. This offset or orientation shall not duplicate the exact meaning of the location or orientation information within the Entity State PDU to which the VP record belongs. The offset is typically between two associated entities, but may also only pertain to the specified entity (such as center of gravity).

The orientation is typically of an associated entity, where one entity will redefine the orientation of another. The orientation assigned by the carrier is an offset of the mounted entity’s orientation in radians when using the Offset Type of Orientation Modified by Carrier (3). Note that the record still uses an Euler Angles record (there is no specific record in IEEE Std 1278.1 that expresses angular offsets). In this case, use psi as the heading offset, theta as the pitch offset, and phi as the roll offset. This matches the descriptions of psi, theta, and phi in IEEE Std 1278.1-2012.

#### Record Definition

The Entity Offset VP record shall contain the following fields:

1. **Record Type**. This 8-bit enumeration shall identify the record as an Entity Offset VP record.
2. **Offset Type**. This 8-bit enumeration shall identify additional details about the type of offset, such as whether it is a positional offset from the carrier origin, station location, or modified by the carrier.
3. **Entity Position Offset / Orientation**. This field shall identify the offset. It shall be represented by an Entity Coordinate Vector record for Entity Position Offset or by an Euler Angles record for Orientation.

Table ‑. Entity Offset VP Record

| Field | Type |
| --- | --- |
| Record Type | 8-bit enumeration |
| Offset Type | 8-bit enumeration |
| Padding | 16-bits unused |
| Entity Position Offset / Orientation | 96-bit record |

#### Issuance Rules

The following issuance rules apply:

1. Multiple Entity Offset VP records may be transmitted per entity, but they shall follow the Entity Association VP record they enhance (prior to any additional Entity Association VP records), if they enhance an Entity Association VP record. Offset Types of 99 or less enhance an Entity Association VP record. Multiple Entity Offset VP records are possible for a single association when the Offset Type is different, such as a positional offset and an orientation.
2. Only one Entity Offset VP record with an Offset Type of Center of Gravity (100) shall be present in one PDU.
3. Once this record is transmitted, it shall only be discontinued when the association no longer exists.
4. The final issuance of this record shall be the one just before disconnecting (i.e., this record is not required in the issuance of the Entity State PDU that indicates the disassociation).
5. If the carrier produces this record, then the Offset Type shall be set to one of the following:
   1. Station Location (1) – the given offset is added to the station position, which is offset from the carrier’s origin; the mounted entity may also modify the offset by providing a Carrier Origin (0) offset
   2. Position Modified by Carrier (2) – if the mounted entity defines a Station Location (1) offset, then this offset is added to that calculated value to determine the final position; otherwise the given offset is added to the carrier’s origin to produce an origin for the mounted entity (and the mounted entity may also modify the offset by providing a Carrier Origin (0) offset)
   3. Orientation Modified by Carrier (3)
6. If the mounted entity produces this record, then the Offset Type shall be set to one of the following:
   1. Carrier Origin (0) – the default value for the Entity Coordinate Vector record; the given offset is added to the position calculated by the carrier origin and any offset supplied by the carrier
   2. Station Location (1) – the given offset is added to the station position, which is offset from the carrier’s origin
   3. Center of Gravity (100) – the given offset is an offset from the entity origin to the center of gravity of the entity

NOTE—The station offset from the carrier’s origin is outside the scope of this specification.

#### Receipt Rules

There are no specific receipt rules for this VP record.

### Dead Reckoning VP record

A side effect of the Entity Offset VP record is that the associated entity (e.g., mounted or towed entity) may not be dead reckoned properly (if its actual dead reckoning parameters are not equivalent to the carrier). Thus, we provide the following Dead Reckoning VP record for linear velocity, linear acceleration, and angular velocity.

#### Purpose

Dead reckoning parameters for other entities shall be communicated with the Dead Reckoning VP record.

#### Record Definition

The Dead Reckoning VP record shall contain the following fields:

1. **Record Type**. This 8-bit enumeration shall identify the record as an Dead Reckoning VP record.
2. **Dead Reckoning Type**. This 8-bit enumeration shall identify the type of dead reckoning parameter included within this VP record.
3. **Dead Reckoning Algorithm**. This 8-bit enumeration shall specify the dead reckoning algorithm.
4. **Dead Reckoning Parameter**. This field shall identify the dead reckoning parameter. It shall be represented by a Linear Velocity Vector record, Linear Acceleration Vector record, or Angular Velocity Vector record.

Table ‑. Dead Reckoning VP Record

| Field | Type |
| --- | --- |
| Record Type | 8-bit enumeration |
| Dead Reckoning Type | 8-bit enumeration |
| Padding | 8 bits unused |
| Dead Reckoning Algorithm | 8-bit enumeration |
| Dead Reckoning Parameter | 96-bit record |

#### Issuance Rules

The following issuance rules apply:

1. Multiple Dead Reckoning VP records may be transmitted per entity, but each one must be with respect to a different parameter (e.g., velocity and acceleration) or a different offset or physical association (such as multiple sling loads).
2. Dead Reckoning VP records shall follow the Entity Offset VP record they extend. For example, a linear velocity Dead Reckoning VP record shall be preceded by a positional Entity Offset VP record, which of course shall in turn be preceded by an Entity Association VP record.
3. Once this record is transmitted, it shall only be discontinued when the association no longer exists or the Dead Reckoning Algorithm changes to a lower fidelity level (e.g., velocity is no longer required).
4. The simulation application shall use the same issuance rules for normal dead reckoning of entity states. For example, if the associated entity (e.g., payload) breaks the dead reckoning threshold but the carrier doesn’t, a new Entity State PDU shall still be issued.
5. If any Dead Reckoning VP records are transmitted, then all required parameters for a given Dead Reckoning Algorithm shall be transmitted. For example, if the dead reckoning algorithm for a sling load payload is listed as FVW (5), then both linear velocity and linear acceleration shall be included.

#### Receipt Rules

There are no specific receipt rules for this VP record.

## Weapons and Detonations

### Weapons Fire

The VDIS recommendation for non-directed energy weapons fire is consistent with the DIS Fire PDU.

### Detonations

The VDIS recommendation for detonations is consistent with the DIS Detonation PDU. VDIS has not yet defined any VP records specifically for the Detonation PDU.

## Collisions

VDIS recommends the usage of the Collision PDU to communicate inelastic or low-fidelity collisions between entities or entities and other objects. VDIS neither recommends nor forbids the usage of the Collision-Elastic PDU to communicate high-fidelity elastic collisions between entities or entities and other objects. We have not established a use case to warrant further investigation of the Collision-Elastic PDU.

VDIS modifies the issuance rules for the Collision PDU that allows for a “third party” simulation application to issue the PDU. In the following cases, the Collision PDU may be issued by a simulation application that is not the issuing entity for the Issuing Entity ID identified in the packet.

1. An RWA that has a sling loaded entity. The RWA models the payload’s position and orientation and includes them in its Entity State PDU. It also therefore performs the collision detection with itself, other entities, and the environment.
2. Any entity that has other entities mounted on it, where the mounted entity position is relative to the carrier. An example of this would be crew members aboard an aircraft carrier that might collide with approaching aircraft.

The Event ID of the Collision PDU shall use the Site Number and Application Number of the issuing simulation application, even if that application is different than the application simulating the Issuing Entity ID. If a third party simulation application outputs a Collision PDU for a collision between itself and that other entity, then only one Collision PDU shall be transmitted for that specific collision instance. If a collision occurs between an outside force (terrain or other entities) and the main entity plus one or more sling loads or mounted entities, then a separate Collision PDU shall be transmitted for the main entity and each, separate sling load or affected mounted entity. For example, if the RWA carrying the sling load crashes into the ground, then at least two Collision PDUs shall be generated, one for the RWA itself and one for the payload (additional ones could also be generated if the crew are modeled distinctly as mounted entities).

VDIS has modified the Collision PDU to utilize the 8-bits of padding to identify the origin of the collisions. This will allow the simulation application to identify if the collision is on the entity itself, on an attached part, articulated part, or one or more of its carried/mounted objects (see section 4.14 for more information on entity associations). This field is now the Collision Origin field and is an 8-bit enumeration. The enumeration as currently defined follows:

Table ‑. Collision Origin Enumeration of the Collision PDU

| Value | Description |
| --- | --- |
| 0 | No Statement |
| 1 | Collision with Attached Part |
| 2 | Collision with Articulated Part |
| 3 | Collision with Mounted Entity |
| 4 | Collision with Sling Load |

The normal operation, which is backward compatible, should be to set the enumeration to Other (0). This implies that the collision occurred with the main body of the entity and that the Site/Application of the Event ID is equal to the Site/Application of the Issuing Entity ID. It may also have occurred with an attached / articulated part for a legacy system that does not support this field. The other enumerated values may indicate that the Site/Application of the Event ID is not equal to the Site/Application of the Issuing Entity ID, such as for a VDIS sling load operation when the payload collides with another entity or the environment. If a collision occurs with part of the issuing entity that is both an articulated and an attached part, then preference in the enumeration should be given to the attached part. If a collision occurs with multiple aspects of an entity (e.g., the main body plus one or more parts), then preference should be given in the order of the enumeration, specifically using Other (0) if the main body is at all impacted. If a collision occurs with the entity and one or more sling loads or mounted entities, then as noted previously, multiple Collision PDUs should be generated.

## Entity Status

### PDU Usage

Entity Status is provided largely from monitoring the Warfare and Entity Interaction families of PDUs. The Entity State PDU provides location, orientation, rate of movement, and appearance data. The entity status data provided in the Entity State PDU is not extensive due to the constraint on the size of the PDU. Additional detail is being provided in Variable Parameter records. The Entity Association VP record provides status of physical linkage or functional associations between two simulated entities or a simulated entity and a simulated object. For a description of the Entity Association VP record, see section 4.14. VDIS defines extensions to the entity appearance data with the use of the VP records of the Entity State PDU. These extensions will provide additional appearance data above what is currently provided.

In addition to the Entity State PDU, the Fire and Detonation PDUs can be monitored to determine ammunition expenditure and fire control.

The Event Report and Data PDUs will be used in VDIS to provide additional details not issued in any of the Entity Interaction and Warfare PDU families (see section 5.5.10 for more information). A Data Query PDU requesting simulation application status related to the current status of entity sensors, entity fuel, munitions and supply levels would provide a simulation application like an IOS or an AAR current manned simulator exercise state. A periodic Data Query request to all manned simulation applications or an unsolicited Data PDU (possibly prompted by a single data query with a recurring update) issued by all manned simulation applications would support data collection during the exercise that could be used post exercise for analysis and to support some level of manned simulation application status during exercise playback.

### Entity Status Datum IDs

The following Variable Datum records will be used in conjunction with the Data Query and Data PDUs to provide entity status. These records may also be used in Event Report PDUs.

Table ‑. Entity Status Datums

| Datum ID | Entity Status Data | Datum Type | Length | Limits/Range/Ref/Data Type |
| --- | --- | --- | --- | --- |
| 500001 | Munition Record | Variable | 128-bits | (See IEEE 1278.1-2012) |
| 500002 | Engine Fuel Record | Variable | 64-bits | (See IEEE 1278.1-2012) |
| 500003 | Storage Fuel Record | Variable | 64-bits | (See IEEE 1278.1-2012) |
| 500005 | Expendable Record | Variable | 128-bits | (See IEEE 1278.1-2012) |
| 500007 | Launched Munition Record | Variable | 384-bits | (See IEEE 1278.1-2012) |
| 500008 | Association Record | Variable | 256-bits | (See IEEE 1278.1-2012) |
| 500009 | Sensor Record | Variable | 128-bits | (See IEEE 1278.1-2012) |

## Sling Loads

### Introduction

The requirements for sling load operations are defined in this section. Sling loads are typically carried by rotary-winged aircraft (RWA) and consist of the hooks, lines, and payload. The payload can be a large number of things, including crates, barrels, and other supplies, stripped down helicopters, land vehicles, and even howitzers. These payloads may be utilized in the simulation purely as loads or may be converted during the exercise to active entities, controlled by SAF operators or even virtual manned modules.

For convenience, this section will refer to the entity carrying the sling load as the “carrier.” The entity or object being carried will be referred to as the “payload.” General requirements for all sling load operations are defined in 4.19.2. Specific requirements to capability requests, attach requests, and detach requests in are defined in following sections.

### General Requirements

The following general requirements shall apply to all sling load operations:

1. The carrier shall set the Sling Load Carrier Capability bit to true (1).
2. Any entity capable of being sling loaded shall set the Sling Loadable Capability bit to true (1).
3. Either the carrier or the payload may initiate the sling load operation: capability request, attach, and detach transactions.
4. The response to any sling load Action Request PDU shall be performed with the Action Response PDU.
5. The Action Response PDU used in sling load operations may contain datum records, but none are currently defined for usage.
6. The Action Response PDU shall be issued within 10 seconds of receipt of the Action Request PDU.
7. The application sending the request shall timeout the request if a response is not provided in time.
8. The sending application shall reset this timer whenever an Action Response PDU with a Request Status of Pending (1) or Executing (2) is received.
9. If the time to process an Action Request PDU is expected to take longer than 10 seconds, then the receiving application shall issue an interim Action Response PDU with a Request Status of Executing (1).
10. If a “hookman” entity (life form) is present, then the Sling Load Pendant flag within the Extended Equipment field for the Legacy Extended Life Form Appearance VP record should be set to Has (1). If this flag is not set, then the hookman will not be able to visualized properly.

### Capability Requests

The following requirements shall apply to all sling load capability requests:

1. Sling load capability may be requested by either the carrier or payload.
2. The capability request shall be performed by issuing an Action Request PDU with an Action ID of Sling Load Capability Request (4300) and using the Sling Load Capability (20030) datum record.
3. The Request Status field in the Action Response PDU should be set to Executing (2), Complete (4), or Request Rejected (5).

### Attachment

The following requirements shall apply to all sling load attachment requests and following successful attachment:

1. Sling load capability may be requested by either the carrier or payload.
2. The attachment request shall be performed by issuing an Action Request PDU with an Action ID of Sling Load Attach Request (4301) and using the Sling Load Capability (20030) datum record.
3. The Request Status field in the Action Response PDU should be set to Pending (1), Executing (2), Complete (4), or Request Rejected (5). The Pending (1) status should be used by the RWA in the case where the payload cannot currently be picked up, but that the RWA is in the process of picking it up (perhaps in the middle of a mission behavior or maneuvering into position). This likely would not be possible for a manned module RWA, but it might function well for SAF RWA where the missions may be preplanned.
4. Upon a successful connection, the following additional rules apply if the simulation application simulating the respective entities supports the Extended Platform Appearance VP record:
   1. The carrier shall modify the Extended Equipment record, Sling Load Status field from Empty (1) to Loaded (2).
   2. If the payload is an entity, then it shall modify the Extended Equipment record, Sling Loaded field from Not Sling Loaded (0) to Sling Loaded (1).
   3. If the payload is an entity, then it shall modify the Status record, Present Domain field to Air (2).
   4. The carrier shall identify sling damage in the Extended Equipment record, Sling Load Damage field.
5. Upon a successful connection, the following additional rules apply:
   1. The carrier shall start transmitting an Entity Association VP record in its Entity State PDU. This record shall set the following fields accordingly:
      1. The Association Status shall be Physical Association Carrier (4).
      2. The Association Type shall be Towed in Air, with the exact enumeration depending on the Hook Type in the Sling Load Capability (20030) datum record, as follows:
         1. Towed in Air (Single Hook, Not Specified) (1) – The hook type is unknown or there is only one possible hook.
         2. Towed in Air (Center Hook) (8) – The hook type identifies the center hook, which is typically out of a possible three hooks.
         3. Towed in Air (Forward Hook) (9) – The hook type identifies the forward hook, which is typically out of a possible three hooks.
         4. Towed in Air (Aft Hook) (10) – The hook type identifies the aft hook, which is typically out of a possible three hooks.
         5. Towed in Air (Tandem Hook) (11) – The hook type identifies two hooks in a correctly managed tandem, which are typically the fore and aft hooks out of a possible three hooks. The center hook is typically unused when a correctly managed tandem hook is indicated.
         6. Towed in Air (Mismanaged Tandem - Fore and Center) (12) – The hook type identifies two hooks in a mismanaged tandem, fore and center hooks
         7. Towed in Air (Mismanaged Tandem - Center and Aft) (13) – The hook type identifies two hooks in a mismanaged tandem, center and aft hooks
         8. Towed in Air (All Hooks) (14) – The hook type identifies all hooks
      3. The Connection Type shall be set appropriately for the sling load type, for example Cable Line (7).
      4. The Associated Object-2 Identifier shall specify the payload.
   2. The payload shall start transmitting an Entity Association VP record in its Entity State PDU. This record shall set the following fields accordingly:
      1. The Association Status shall be Physical Association Target (1).
      2. The Association Type shall be the same as specified by the carrier.
      3. The Connection Type shall be the same as specified by the carrier.
      4. The Associated Object-2 Identifier shall specify the carrier.
   3. The carrier may use only one Entity Association VP record for the entire sling load operation, even if multiple physical lines are attached.
   4. *Position and Orientation*.
      1. The carrier shall transmit one or more Entity Offset VP records following the Entity Association VP record. One of these records shall denote the payload position. Another may be issued to denote orientation of the payload.
      2. Receiving applications shall use the position (and possibly orientation) as specified or modified by the carrier. Typically, this will mean that the position in the payload Entity State PDU is not used.
      3. The payload entity shall update its correct position and orientation, in order to support legacy systems that do not process the Entity Offset VP record.
   5. *Dead Reckoning.*
      1. The carrier should transmit one or more Dead Reckoning VP records to provide high fidelity dead reckoning parameters for the payload position.
      2. If these Dead Reckoning VP records are not provided, then receiving simulation applications should use the dead reckoning parameters of the carrier to update the position of the payload, or dead reckon the carrier before applying the position offset.
      3. If a receiving simulation does not receive all of the required dead reckoning parameters, then it shall use the carrier’s dead reckoning parameter for those that are missing. For example, if the dead reckoning algorithm for a payload is listed as FVW (5), and only linear velocity and the offset are included in VP records, then the linear acceleration of the carrier is also used to properly dead reckon the payload.
      4. The payload entity shall still provide dead reckoning parameters in its Entity State PDU, in order to support legacy systems that do not process the Dead Reckoning VP record.
      5. The carrier that issues these dead reckoning parameters shall use the same issuance rules for normal dead reckoning of entity states. For example, if the payload breaks a dead reckoning threshold, but the carrier doesn’t, a new Entity State PDU shall still be issued.
6. The following additional rules apply with respect to damage:
   1. The payload shall continue to model battle and collision damage for itself, as it normally would.
   2. The carrier shall model battle and collision damage for the line and sling, if line and sling damage are modeled. See d) 4).
7. The following additional rules apply with respect to collisions, if collision detection is supported by the simulation application simulating the carrier:
   1. The carrier shall perform collision detection for the line(s), sling, and payload.
   2. The carrier shall output the Collision PDU for itself, line(s), sling, or payload when a collision is detected with itself, another entity, another object, or the terrain.
   3. When the carrier collides with the payload it is carrying, the carrier shall output only one Collision PDU with the Collision Origin field set to Collision with sling load (4). No second Collision PDU shall be transmitted.
   4. When the carrier outputs the Collision PDU for the payload, then it shall set the Issuing Entity ID to the payload Entity ID and use an Event ID with itself as the Site and Application numbers.

### Detachment

The following requirements shall apply to all sling load detachment requests and following successful detachment:

1. Sling load detachment may be requested by either the carrier or payload.
2. The detachment request shall be performed by issuing an Action Request PDU with an Action ID of Sling Load Detach Request (4302).
3. The Request Status field in the Action Response PDU should be set to Executing (2), Complete (4), or Request Rejected (5).
4. Upon successful detachment, the following additional rules apply:
   1. The carrier shall send out a revised Entity State PDU as follows:
      1. It shall include a final Entity Association VP record with an Association Status set to Association Broken (3).
      2. It shall no longer include Entity Offset VP or Dead Reckoning VP records (for this sling load operation).
   2. The payload shall send out a revised Entity State PDU to include a final Entity Association VP record with an Association Status set to Association Broken (3).
   3. The carrier shall no longer perform collision detection for the payload.
5. Upon successful detachment, the following additional rules apply if the simulation application simulating the respective entities supports the Extended Platform Appearance VP record:
   1. The carrier shall modify the Extended Equipment record, Sling Load Status field from Loaded (2) to Empty (1).
   2. If the payload is an entity, then it shall modify the Extended Equipment record, Sling Loaded field from Sling loaded (1) to Not sling loaded (0).
   3. If the payload is an entity, then it shall modify the Status record, Present Domain field to the current domain (where it was dropped), which is likely Land (1) or Unchanged (0).
   4. The carrier shall continue identifying sling damage in the Extended Equipment record, Sling Load Damage field.

## Life Form Parachuting

This section summarizes implementation details for a paratrooper progressing through a parachuting sequence. The VDIS recommendation for a paratrooper uses several variable parameter records and the appearance record of the entity state PDU for the paratrooper.

* Life form entity type: use “legacy” enumerations, so based on Air domain:
  + E.g., 3.2.225.1.32.1.0 – U.S. Paratrooper, M16A2
  + E.g., 3.2.225.1.41.1.0 – U.S. Paratrooper, M4 Carbine
* “Parachute, deployed, on ground” entity type: 6.0.0.0.7.6.1
* Define, but do not use other parachute types:
  + Parachute: 6.0.0.0.7.6.0
  + Parachute, deployed, in tree: 6.0.0.0.7.6.2

### Visualization

A life form entity may be visualized while parachuting using entity association, mounted status and posture. If ground clamping is applied to the parachuting life form, then present domain is needed to indicate that the life form is not yet on the ground so as to suppress ground clamping during descent. The visual states for parachuting are mounted in aircraft, freefall descent, deployed descent, first surface contact, chute detached.

### Mounted on Aircraft

When the life form is mounted on the plane:

* Entity State PDU  
  🡪Appearance🡪Life Form State (Posture, bits 16-19) = as normal, e.g., Upright, standing still (1)  
  🡪Appearance🡪Mounted (bit 22) = True (1)  
  🡪Legacy Extended Life Form Appearance VP record🡪Status🡪Present Domain = Interior Other Entity/Object (12)
* Include the Entity Association VP record as normal for mounting operations (both the life form and the aircraft)

### Freefall Descent (OPTIONAL)

When the life form has dismounted and before deploying the parachute (freefall):

* Entity State PDU  
  🡪Appearance🡪Life Form State (Posture, bits 16-19) = as normal, e.g., Prone (5)  
  🡪Appearance🡪Mounted (bit 22) = False (0)  
  🡪Legacy Extended Life Form Appearance VP record🡪Status🡪Present Domain = Air (2)

### Deployed Descent

When the life form has deployed the parachute:

* Entity State PDU  
  🡪Appearance🡪Life Form State (Posture, bits 16-19) = Parachuting (8)  
  🡪Appearance🡪Mounted (bit 22) = False (0)  
  🡪Legacy Extended Life Form Appearance VP record🡪Status🡪Present Domain = Air (2)
* Remove the Entity Association VP record (both the life form and the aircraft).

### First Surface Contact

When the life form has first landed:

* Entity State PDU  
  🡪Appearance🡪Life Form State (Posture, bits 16-19) = Prone (5)  
  🡪Legacy Extended Life Form Appearance VP record🡪Status🡪Present Domain = Land (1)

### Chute Detached

When the life form stands up, after landing:

* Entity State PDU  
  🡪Appearance🡪Life Form State (Posture, bits 16-19) = Upright, standing still (1)  
  🡪Legacy Extended Life Form Appearance VP record🡪Status🡪Present Domain = Land (1)
* Create a new “Parachute, deployed, on ground” entity

### Known issues

* This does not distinguish whether the paratrooper still has a parachute attached or not while on the ground, including whether or not said parachute is deployed. If this level of fidelity is required, there’s already a field in VDIS to support it ( Entity State PDU🡪Extended Life Form Appearance VP record🡪Extended Equipment🡪Parachute (bits 1-2) )
* A Present Domain of “Unchanged” is not recommended for legacy life forms because the domain is Air (2).

# SIMAN

The VDIS specification supports multiple concurrent exercises on a single DIS network. This follows the DIS standard with the exercise number specified in the PDU header. To facilitate the control of system assets and their assignment to various exercises from a central location at a site, VDIS reserves Exercise 0 (NO\_EXERCISE, see section 3.1.2.2) to support scenario initialization, asset control and exercise management protocols. VDIS provides for the dynamic discovery of assets as they come on-line using the same methodology for entities (namely listening for new heartbeats). For the following sections, VDIS also recommends the usage of an Asset Manager and Exercise Manager. These systems are simply a differentiation of functionality and thus all simulation management functions could be (and generally are) performed by the same simulation application. Consequently, these terms may be used interchangeably if it fits a training system’s design and we’ll use the term Simulation Manager to represent any and both of them.

## Reliability

VDIS recommends the usage of the Simulation Management family instead of the Simulation Management with Reliability family of PDUs. The inclusion of a single field to add the capability of application-level reliability is valuable, but necessary compared with the significant level of changes required for legacy systems. The following table otherwise specifies a comparison of Simulation Management PDUs with Simulation Management with Reliability PDUs. It’s important to note that Application Control is not actually currently part of either family (it’s in the experimental family).

Table ‑. Comparison of Reliability PDUs

| Simulation Management |  | Simulation Management w/ Reliability |
| --- | --- | --- |
| Start/Resume | ↔ | Start/Resume-R |
| Stop/Freeze | ↔ | Stop/Freeze-R |
| Create/Remove Entity | ↔ | Create/Remove Entity-R |
| Action Request | ↔ | Action Request-R |
| Action Response | ↔ | Action Response-R |
| Acknowledge | ↔ | Acknowledge-R |
| Data Query | ↔ | Data Query-R |
| Set Data | ↔ | Set Data-R |
| Data | ↔ | Data-R |
| Event Report | ↔ | Event Report-R |
| Comment | ↔ | Comment-R |
| Application Control | ↔ | Application Control |

Action Request and Action Response are discussed in section 5.1.1. Data Query, Set Data, and Data are discussed in section 5.1.2. Start/Resume and Stop/Freeze are discussed in sections 5.5.3 and 5.5.4, respectively. Event Report is discussed in section 5.5.10. Comment is discussed in section 5.5.11. The creation and removal of entities is discussed in section 4.1. VDIS does not currently have a recommendation for the usage of Record Query-R, Set Record-R, or Record-R. The Application Control PDU is discussed in section 5.2.

### Action Request / Action Response

The Simulation Manager or a simulation application utilizes Action Request/Action Response PDUs in order to request that another simulation application perform a specific action. The simulation manager or a simulation application initiates the protocol by sending the receiving simulation application an Action Request PDU with a requested action identifier. The simulation application responds to the Action Request PDU by sending an Action Response PDU within ten (10) seconds of the Action Request PDU time stamp. The simulation application specifies the current request status when it sends the Action Response PDU.

While the simulation application is processing the request, the simulation manager (or sending simulation application) may periodically ask for the status of the original request by retransmitting the original Action Request PDU (datum values do not need to be included if the original Action Request was acknowledged). The request identifier in the Action Request PDU is used to correlate the status request to the original request. The simulation application responds to the status request by sending an Action Response PDU with the current request status within ten (10) seconds of the latest Action Request PDU time stamp. The simulation application sends an Action Response PDU with a complete status to the simulation manager (or sending simulation application) when the application has completed the requested action.

### Data Query / Set Data / Data

A request for data shall be communicated by issuing a Data Query PDU.

The receiving simulation application of a Data Query PDU shall respond by issuing a Data PDU containing the values of the datum records requested to which the application is capable of replying. Multiple Data PDUs may be sent in response to a single Data Query. The receiving application may split up the data for purposes of PDU size, execution time (to fulfill the request), or some other reason. The request is not considered fulfilled until all datum IDs are received by the requesting simulation application.

The VDIS protocol interactions for the Data Query PDU are otherwise consistent with the interactions described in DIS. The response Data PDU shall be sent on the same network (and exercise) as the Data Query PDU.

Initializing or changing internal state information shall be communicated using a Set Data PDU. The Set Data PDU may be used to change internal state data. The VDIS protocol interactions for the Set Data PDU are consistent with the interactions described in DIS. The response Data PDU will be sent on the same network (and exercise) as the Set Data PDU.

Information issued in response to a Data Query PDU or Set Data PDU shall be communicated using a Data PDU. A Data PDU may also be used as an unsolicited transmission, but not for Event Reporting (see section 5.5.10). The VDIS usage of the Data PDU is otherwise consistent with the usage indicated in DIS.

## Application Control PDU

### Purpose

The Application Control PDU shall be used by a Simulation Manager (SM) to attempt control of an application, when data is required from an application, or when an application is required to change its parameters. The data required is typically for the express purpose of subsequently controlling the application.

### Information contained in the Application Control PDU

The Application Control PDU shall contain the following information:

1. Simulation Management PDU Header
2. Level of reliability service to be used for the requested action
3. Request identification number for the action being requested by the Simulation Manager (SM)
4. Identification of the specific action to be taken
5. Data required for performing the requested action

### Issuance of the Application Control PDU

The following issuance rules apply:

1. The Application Control PDU shall be issued using a best effort unicast or multicast communication service.
2. The Application control PDU shall be issued by a simulation application in response to another Application Control PDU, as indicated in the specific issuance rules for those records.
3. The Application Control PDU should be issued on Exercise 0. It may be issued on particular exercises for certain functions as defined in the specific interaction rules for those functions or records.
4. If a control message is too big to fit into one PDU due to network MTU size issues (or other parameters as defined by the simulation application or per exercise agreement), the Number of Parts and Current Part fields shall be used to indicate that the entire message spans multiple PDUs.

The following specific field requirements apply:

1. An Application Control PDU with a Control Type of Data Query (2) shall use the acknowledge reliability service level.
2. A Control Type of Status (7) shall not be issued using the acknowledge reliability service level.
3. If a Control Type of Data Query (2) is sent, then the Application Control PDU should contain a single record.
4. The difference between the Control Type of Set Data (4) and Add Data (5) relates to the type of data being updated. If the data being updated contains multiple records and the complete set of records is being sent, the Control Type shall be Set Data (4). If the data being updated contains multiple records and only new records are being added, the Control Type shall be Add Data (5).
5. A Time Interval of 0 seconds shall signify an instantaneous request and does not “turn off” heartbeats.
6. A Time Interval of 255 shall signify disabling heartbeats, but this should be used very carefully and only under specific exercise agreement.
7. If the control message is contained within a single PDU (regardless of the number of records) then the Number of Parts shall be set to 1.
8. The maximum value for Number of Parts is 255.
9. If no records are present for an issuance of the PDU, the Number of Application Control Records field shall be set to zero and no records shall be included.

### Receipt of the Application Control PDU

The following receipt rules apply:

1. Response Application Control PDUs shall be issued on the same exercise ID as the PDU responded to, or Exercise 0 if all exercises were originally specified.
2. Response Application Control PDUs shall never be transmitted on All Exercises (255).
3. Response Application Control PDUs shall contain the same Request ID as the corresponding request PDUs and the Required Reliability Service shall be set to Unacknowledged (1).
4. A response may consist of multiple Application Control PDUs instead of a single PDU. Responses that span multiple PDUs shall have the Current Part and Number of Parts fields set appropriately.
5. If an Application Control PDU with a Control Type of Shutdown (1) indicates that the acknowledge reliability service level shall be used for a transaction, an Application Control PDU with a Control Type of Other (0) shall be issued upon receipt of the PDU. The reply may or may not contain any parameter values.
6. A simulation application shall respond with an Application Control PDU with a Control Type of Data (3) upon receipt of the PDU. When the response PDU is issued, it shall contain the value or values (if any) of the requested record types to which the application is capable of replying.
7. If an Application Control PDU with a Control Type of Set Data (4) or Add Data (5) indicates that the acknowledge reliability service level shall be used for a transaction, an Application Control PDU with a Control Type of Data (3) shall be issued upon receipt of the PDU. When the response PDU is issued, it shall verify the receipt of the PDU by returning the parameter values that were set in response to the records sent. Parameters that were set in the simulation to the same values as in the request PDU shall be set to those values in the response PDU. Parameter values that were set to different values in the simulation than requested shall be set to their actual values in the response PDU. Parameters to which the receiving application cannot comply shall not be included in the response PDU. The response PDU shall contain similar information as to that which would be sent in response to a Control Type of Data Query (2).
8. If an Application Control PDU with a Control Type of Data (3) indicates that the acknowledge reliability service level shall be used for a transaction, an Application Control PDU with a Control Type of Data (3) shall be issued upon receipt of the PDU. The reply shall not contain any parameter values.
9. If an Application Control PDU with a Control Type of Remove Data (6) indicates that the acknowledge reliability service level shall be used for a transaction, an Application Control PDU with a Control Type of Data (3) shall be issued upon receipt of the PDU. When a response PDU is issued, it shall verify the receipt of the PDU by returning the resultant parameter values after the removal of the specified records. Parameters to which the receiving application cannot comply shall not be included in the response PDU. The response PDU shall contain similar information as to that which would be sent in response to a Control Type of Data Query (2).

NOTE—The rules on issuance of the Application Control PDU also apply to response PDUs.

### PDU Definition

Control of an application shall be communicated through the Application Control PDU. The Application Control PDU shall consist of the following fields:

1. **Simulation Management PDU Header**. The Simulation Management PDU Header shall be represented by the Simulation Management PDU Header record. It consists of the PDU Header, the Originating ID and the Receiving ID.
2. **Required Reliability Service**. This 8-bit enumeration shall identify the level of reliability service to be used for this transaction.
3. **Time Interval**. This 8-bit unsigned integer shall identify the time interval (heartbeat) in seconds for the requested Control Type. This field is only applicable to specific Control Types as identified in the sequences in the following SIMAN sections (e.g., Status for Application State under Asset Control).
4. **Control Type**. This 8-bit enumeration shall specify the particular control action that is being requested.
5. **Originating Application Type**. This 16-bit enumeration shall specify the type of application originating this PDU.
6. **Receiving Application Type**. This 16-bit enumeration shall specify the type of application intended to receive this PDU.
7. **Request ID**. This 32-bit unsigned integer shall uniquely identify the request being made by the issuing application.
8. **Number of Parts**. This 8-bit unsigned integer shall specify the total number of PDUs that this control message is sub-divided into.
9. **Current Part**. This 8-bit unsigned integer shall specify the current part out of the **Number of Parts**. The parts are numbered uniquely from 1 to 255.
10. **Number of Application Control Records**. This 16-bit unsigned integer shall identify the number of Application Control records.
11. **Application Control records**. These fields shall contain one or more Application Control records and may contain other Standard Variable records. These records shall conform to the variable record format of a Standard Variable Specification record.

The format of the Application Control PDU shall be as shown in Table 5‑2.

Table ‑. Application Control PDU

| Field Size  (bits) | Application Control PDU Fields | |
| --- | --- | --- |
| 96 | PDU Header | Protocol Version - 8-bit enumeration |
| Exercise ID - 8-bit unsigned integer |
| PDU Type - 8-bit enumeration |
| Protocol Family - 8-bit enumeration |
| Time Stamp - 32-bit unsigned integer |
| Length - 16-bit unsigned integer |
| PDU Status - 8-bit record |
| Padding - 8-bits unused |
| 48 | Originating ID | Site Number - 16-bit unsigned integer |
| Application Number - 16-bit unsigned integer |
| Reference Number - 16-bit unsigned integer |
| 48 | Receiving ID | Site Number - 16-bit unsigned integer |
| Application Number - 16-bit unsigned integer |
| Reference Number - 16-bit unsigned integer |
| 8 | Required Reliability Service | 8-bit enumeration |
| 8 | Time Interval | 8-bit unsigned integer |
| 8 | Control Type | 8-bit enumeration |
| 8 | Padding | 8-bits unused |
| 16 | Originating Application Type | 16-bit enumeration |
| 16 | Receiving Application Type | 16-bit enumeration |
| 32 | Request ID | 32-bit unsigned integer |
| 8 | Number of Parts | 8-bit unsigned integer |
| 8 | Current Part | 8-bit unsigned integer |
| 16 | Number of Application Control Records (*N*) | 16-bit unsigned integer |
| varies | Application Control record #*1* | Record Type – 32-bit enumeration |
| Record Length – 16-bit unsigned integer (6+K*1*+P*1*) |
| Record-Specific Fields – K*1* octets |
| Padding to 64 bits – P*1* octets |
|  | … |  |
| varies | Application Control record #*N* | Record Type – 32-bit enumeration |
| Record Length – 16-bit unsigned integer (6+K*N*+P*N*) |
| Record-Specific Fields – K*N* octets |
| Padding to 64 bits – P*N* octets |
| Total Application Control PDU size =  bits  The Record Length (6+K*i*+P*i*) value shall be a multiple of 8 octets. | | |

### Application Control Enumerations

The Control Type enumeration follows:

Table ‑. Control Type Enumeration for Application Control PDU

| Value | Description |
| --- | --- |
| 0 | Other |
| 1 | Shutdown |
| 2 | Data Query |
| 3 | Data |
| 4 | Set Data |
| 5 | Add Data |
| 6 | Remove Data |
| 7 | Status |

The Application Type enumeration follows:

Table ‑. Application Type Enumeration for Application Control PDU

| Value | Description |
| --- | --- |
| 0 | Other |
| 1 | Resource Manager |
| 2 | Simulation Manager |
| 3 | Gateway |
| 4 | Stealth |
| 5 | Tactical Internet Interface |

## Scenario Initialization

VDIS scenario initialization is performed via the Simulation Manager product distributing the scenario initialization data to all simulation applications participating in an exercise. The Simulation Manager provides initialization instructions to each simulation application to read its initialization parameters from its view of the scenario file. The scenario file data will be tagged to identify the initialization data that is appropriate for each simulation application. Dependent on the simulation application, its scenario data will be found in a common scenario file or a scenario file that has been tailored for specific simulation applications. The simulation application will parse its initialization parameters from the scenario file and apply values to initialize the application for scenario start. When the scenario initialization for a simulation application is complete, the simulation application will notify the Simulation Manager that it is ready to start exercise simulation.

### Initialization Files

The SE Core program is working towards the development of a standard exercise file definition. The exercise file shall provide definitions for scenario, configuration, asset assignment, radio networks and other items associated with an exercise or series of exercises. The files shall reside on a shared file system within the training suites or sites so all systems would have access to them. The current VDIS definitions try to reflect these concepts without eliminating existing capabilities. Even if a common file format is not available or possible, the VDIS approach would still work well.

### Initialization Sequence

The VDIS Initialization capability establishes the set of initial conditions for a simulation application to operate under during exercise simulation. Initialization cannot occur until a simulation application has joined an exercise and is therefore listening to that exercise traffic. The simulation application must have a Current Exercise State of Allocated (2) or Initialized (3). In the second case, this sequence would merely cause it to reinitialize. If the simulation application is not in one of the two listed states for the specified exercise, then it shall ignore all attempts at initialization.

The VDIS Simulation Manager shall send an Application Control PDU (Set Data (4), Application Initialization Data Record (47300)) in order to establish a simulation application's set of initial conditions. The Exercise File Path and Exercise File Name provide the location and name of the exercise file to be loaded from the shared file system. The Application Role defines the current role of the simulation application and is used as a label to parse the exercise file to extract the application’s initialization data. The role concept allows for a different asset to be assigned to a different entity within the exercise while also allowing multiple initialization sets that can be dynamically allocated to applications within a single scenario file. This PDU may be transmitted on Exercise 0, but VDIS recommends that it be transmitted on a specific exercise (given prior exercise agreement this may be done in order to reduce traffic on Exercise 0 when there are a lot of assets and concurrent exercises). The data record contains the exercise ID to initialize for convenience depending on which method is chosen.

If acknowledged reliability is requested, then the simulation application shall immediately respond with an Application Control PDU (Data (3), Application Initialization Data Record (47300)) to acknowledge the instruction. The Simulation Manager shall listen for updated Application State (47100) records to determine the status of the simulation application complying with the request to initialize. The Simulation Manager may retransmit the initialization request with the same ID, and, when this is done, the simulation application shall immediately respond as above. The simulation application shows compliance by an immediate change in state, transmitting its heartbeat Application State (47100). See section 5.4.2 for more information on application state. Note that the simulation application has the option to modify the initialization data when it responds to the reliability request. It may opt to do this to signify its preference in a file format, although how that information is useful or used by the Simulation Manager is not currently relevant to the VDIS interaction protocol (e.g., it may be used for error checking).

Figure 5‑1 shows the complete interaction. The number next to the line identifies the recommended exercise. The Simulation Manager can send additional Application Control PDUs (Set Data (4)) or even Set Data PDUs to applications with various entity parameters to alter the definitions as initialized from the scenario file. This allows the operator to alter the scenario prior to or perhaps during the exercise. These additional initialization parameters should be transmitted on the same exercise (either 0 or N) that was used for the Application Initialization Data (47300).



Figure ‑. VDIS Scenario Initialization

## Asset Control

### Asset Discovery

Within the VDIS specification a method of asset discovery has been added. This provides the capability for the Simulation Manager to discover which assets are available as they come online and allows the Simulation Manager to allocate them to an exercise. The Simulation Manager is specific to a given site and should be the first application brought up. The site application manager should start by listening for PDUs on Exercise 0. When an asset instantiates itself, it will send one or more unsolicited, unacknowledged Application Control (Status (7), Application State (47100)) PDUs to the all applications on Exercise 0 showing that it has come online. The asset retransmits these PDUs at the defined heartbeat rate (see section 5.4.2 for more information). The Simulation Manager is able to assign specific roles to assets via the Application Control PDU (Data (3), Application Initialization Data Record (47300)). Please see 5.3 for more information.

For redundancy or ease of use, a given site may have more than one Asset Manager. VDIS supports a peer to peer methodology that allows the Simulation Managers to cooperate with each other. The Simulation Manager may also alter the status heartbeat frequency using the Application Control PDU Time Interval field for the Status (7) Control Type. If the state and status heartbeats become an issue, this could be used to reduce their frequency of update.

### Application State and Capability

VDIS Application State and Capability protocols are illustrated in Figure 5‑2. In order to establish and maintain a connection to the Simulation Manager from a simulation application, VDIS requires that a simulation application periodically transmit its Application State (47100) via the Application Control PDU (Status (7)) on Exercise 0. The heartbeat rate at which the Application State (47100) shall be transmitted is 5 seconds; however, this heartbeat rate can be modified by setting the Time Interval field in the Application Control PDU when sending a Data Query on the Application State record (47100) (see the figure below). The simulation application shall transmit a new state record immediately whenever its state changes, such as when an exercise state change occurs or is occurring. In addition to the heartbeat, the Simulation Manager (or another simulation application) can request an Application State (47100) via the Application Control PDU (Data Query (2)). In that case, the Application State (47100) will be transmitted via a Control Type of Data (3), not Status (7).

During an application state change, the Requested Application State shall indicate the target state and the Current Application State shall be set to Offline/Unknown (0). Additionally, the Application Transition field shall be set to Transitioning (1) during the process and to Not Transitioning (0) when complete. Other applications monitoring the state can then use these values as an indication that the application is processing a change request, or is otherwise in transition. If the state change happens to fail, then the Current Application State shall be set to Error (3) and not revert back to its previous state.



Figure ‑. VDIS Application State and Capability

During an exercise state change, the Requested Exercise State (for a particular exercise per application) shall indicate the target state and the Current Exercise State shall be set to Unknown (0). Additionally, the Exercise Transition field shall be set to Transitioning (1) during the process and to Not Transitioning (0) when complete. Other applications monitoring the state can then use these values as an indication that the application is processing a change request, or is otherwise in transition. If the state change happens to fail, then the Current Exercise State shall be set to Error (6) and not revert back to its previous state.

The simulation application also identifies its current application configuration, which is an index into an application-specific list of configurations. This list can be established *a priori* or queried via the Application Capability (47500) record using an Application Control PDU (Data Query (2)). In fact, the Simulation Manager may perform this query at the beginning of the exercise in order to obtain the other information transmitted in the Application Capability (47500) record (maximum number of supported exercises and application identification information). This record shall be transmitted on Exercise 0.

### Application Health Status

VDIS Application Health Status protocols are illustrated in Figure 5‑3. VDIS requires that a simulation application periodically transmit its Application Health Status (47200) via the Application Control PDU (Status (7)) on Exercise 0 detailing the status of various hardware and software components, as defined by the Simulation Manager (or per initial exercise agreement). The heartbeat rate at which the Application Health Status (47200) must be transmitted is defined per Status Type in the Application Health Status Heartbeat Request record (47240); however the default is that no heartbeat for health status is issued initially unless otherwise specified by exercise agreement. The simulation application shall also transmit a new Application Health Status record (47200) immediately whenever the state of that particular status field changes within some threshold defined by the application and Simulation Manager (for those fields requested by heartbeat or per initial exercise agreement only, see the description on Status Threshold in the Application Health Status Heartbeat Request record (47240) for more details).



Figure ‑. VDIS Application Health Status

The Simulation Manager (or another simulation application) can query a simulation application health status capability via the Application Health Status Capability record (47230). The health status types used in these records are either predefined per exercise agreement or can be queried from the individual applications via the Application Status Type Description record (47210), which will identify the short name (label) and description of the given type. The Simulation Manager should attempt to query the simulation applications health status capability before requested specific status type heartbeats.

The actual health status for each Status Type as given in the Application Health Status record (47200) is identified by the General Status, Specific Status, and Status Value. The General Status is defined in VDIS with the Application General Status enumeration. The Specific Status per each General Status is application specific, although the value for 0 is reserved as “not specified” or its equivalent. The Simulation Manager can query the descriptions of these enumerations from the simulation application via the Application Status Description record (47220).

### Asset Resignation

This section will be completed at a future release of this document.

### Daily Readiness Check

This section will be completed at a future release of this document.

## Exercise Management

### Join Exercise

VDIS Join Exercise protocols are illustrated in Figure 5‑4. To assign an asset to an exercise, the Simulation Manager sends an Action Request (Join Exercise, Exercise ID (47400)) for the Application to join a specific exercise and the asset responds with an Action Response PDU (Executing, Exercise ID (47400)) acknowledging the assignment. Upon a successful transition, the asset then begins listening to the specified exercise. The asset then transmits an Application Control (Status (7), Application State (47100)) PDU on Exercise 0 with the new exercise specified that is picked up by all systems concerned about assets. The asset then issues an Action Response (Complete, Exercise ID (47400)) to the Simulation Manager. Finally, the asset maintains its Application State (47100) heartbeat on Exercise 0 at 5 seconds (unless otherwise instructed with a different Time Interval).



Figure ‑. VDIS Join Exercise

### Resign from Exercise

VDIS Resign from Exercise protocols are illustrated in Figure 5‑5. The VDIS Simulation Manager first stops the simulation application in order to halt all simulated entities controlled by it (see section 5.5.4 for details). The simulation manager then sends an Action Request PDU (Resign Exercise, Exercise ID (47400)) to the simulation application on Exercise 0. The simulation application acknowledges the PDU request by sending an Action Response PDU (Executing, Exercise ID (47400)). The application then informs the Simulation Manager on Exercise 0 that it has resigned from the scenario and is available for use within another exercise. The asset then issues an Action Response (Complete, Exercise ID (47400)) to the Simulation Manager. Finally, the asset maintains its Application State (47100) heartbeat on Exercise 0 at 5 seconds (unless otherwise instructed with a different Time Interval).



Figure ‑. VDIS Resign from Exercise

### Start / Resume

VDIS Start/Resume protocols are illustrated in Figure 5‑6. The VDIS Simulation Manager sends the Start/Resume PDU on the exercise to a simulation application in order to start or resume simulation operations within the specified exercise. The Simulation Application acknowledges the Start/Resume PDU by sending an Acknowledge PDU on the Exercise to the Simulation Manager within ten (10) seconds of the Start/Resume PDU DIS time stamp (and before the time specified in the PDU). The simulation application transitions to the simulating state at the real-world time specified in the Start/Resume PDU. The simulation application sets internal time-of-day parameters, as applicable, based on the simulation time as specified in the Start/Resume PDU. While in the simulating state, the simulation application maintains real-time time-of-day updates. When the simulation application actually changes state, it then updates its Application State (47100) and immediately sends out an Application Control (Status (7)) update, and then resumes normal heartbeats.



Figure ‑. VDIS Start/Resume

### Stop / Freeze

VDIS Stop/Freeze protocols are illustrated in Figure 5‑7. The VDIS Simulation Manager sends the Stop/Freeze PDU on the Exercise to the simulation application in order to pause the simulation application’s simulation operations. The VDIS implementation does not specifically support reason codes indicating the Simulation Manager’s intent. The simulation application acknowledges the Stop/Freeze PDU by sending an Acknowledge PDU on the Exercise to the Simulation Manager within ten (10) seconds of the Stop/Freeze PDU DIS time stamp (and before the time specified in the PDU). The simulation application transitions all simulated entities it is responsible for managing to the Stop/Freeze state in accordance with the Frozen Behavior flag at the real-world time specified in the Stop/Freeze PDU. When the simulation application actually changes state, it then updates its Application State (47100) and immediately sends out an Application Control (Status (7)) update, and then resumes normal heartbeats. The Simulation Manager may at this point alter the Time Interval of the simulation application so that it heartbeats less often. Although VDIS does not recommend disabling the Application State (47100) heartbeat (unless per exercise agreement), using a Time Interval of 255 would disable it.



Figure ‑. VDIS Stop/Freeze

### Restart

This section will be completed at a future release of this document.

### Reconstitute

This section will be completed at a future release of this document.

### Reposition

This section will be completed at a future release of this document.

### Checkpoint / Reset

This section will be completed at a future release of this document.

### Exercise Management

This section will be completed at a future release of this document.

### Event Reporting

The Event Report PDU shall be used to communicate the occurrence of a significant event. The types of significant events recommended for Event Report PDU usage are events that provide additional information to the Simulation Manager for error conditions that must be logged for asset performance analysis and for events that provide additional data to the AAR to supplement the data not provided in normal PDU interactions

The VDIS Event Report PDU shall be issued by a simulation application to a Simulation Manager to communicate that a significant event has occurred relative to the performance of the asset. For significant event data directed to the AAR, the applications shall be configured to send the event to the multicast group allocated to traffic for the exercise in which the AAR is currently recording. The expected Receiving ID for Event Report PDUs is the current site (e.g., 9307), all applications (0xFFFF), 0. Note that for this PDU, using all entities for the last field (0xFFFF) is not allowed per the DIS standard.

Event Report PDUs are issued primarily to provide status to the AAR station. The following table identifies the Variable Datum records used in conjunction with the Event Report PDU to provide entity status to the AAR. See APPENDIX A for the description of the of the DIS datum records that provide entity status to the AAR. VDIS has currently not defined any event reports targeted outside of the AAR.

Table ‑. AAR Event Report Datum IDs

| Datum ID | Description | Datum Type | Length | Limits/Range/Ref/Data Type |
| --- | --- | --- | --- | --- |
| 35000 | Damage Event Report | Variable | 128-bits | (See record defined in appendix) |
| 42050 | Mission Request Event Report | Variable | 128-bits | (See record defined in appendix) |
| 42060 | Indirect or CAS Fire Event Report | Variable | 64-bits | (See record defined in appendix) |
| 42070 | Minefield Entry | Variable | 128-bits | (See record defined in appendix) |
| 42080 | Mine Detonation | Variable | 128-bits | (See record defined in appendix) |
| 35500 | Radar Track Perception Report | Variable | 64-bits | (See record defined in appendix) |

#### Event Reporting Usage

Simulation Applications that model minefields shall issue an Event Report PDU when an entity first passes through an uncleared minefield. This PDU shall have an Event Type of Minefield Entry (11) and contain one datum, the Minefield Entry (42070) Variable Datum record. When the entity subsequently reenters the same minefield, the Event Report PDU is optional.

Simulation applications that model minefields shall issue an Event Report PDU when a mine detonates within a minefield. This PDU shall have an Event Type of Minefield Detonation (12) and contain one datum, the Mine Detonation (42080) Variable Datum record.

Simulation Applications with entities that suffer damage shall issue an Event Report PDU for each incident of combat, deterministic, and stochastic damage. The only exception to this is collision damage, which does not result in the issuance of an Event Report PDU since crew casualties are not assessed for collisions. Instead, the collision damage information is obtained from the next Entity State PDU to be issued by the entities involved in the collision. Since crew casualties are not assessed for collisions and CGF vehicles, the number of casualties shall be assumed to be zero. This Event Report PDU shall have an Event Type of Damage (4) and contain the Damage Event Report (35000) Variable Datum record.

Simulation applications that receive requests for indirect fire missions shall issue an Event Report PDU upon receiving a radioed request for a mission. This PDU shall have an Event Type of Request For Indirect Fire or CAS Mission (9) and contain one datum, the Mission Request Event Report (42050) Variable Datum record.

Simulation applications that issue Fire PDUs for indirect or CAS fire shall also issue an Indirect or CAS Fire Event Report PDU immediately following the Fire PDU. This Event Report PDU associates the Fire PDU with a mission. This PDU shall have an Event Type of Indirect Fire or CAS Fire (10) and contain one datum, the Indirect or CAS Fire Event Report (42060) Variable Datum record.

Simulation Applications that model Radar Warning Receivers shall issue follow the rules for the Radar Track Perception Record as identified in the appendix.

#### Enumerations for Event Report Datums

The following table provides the Damage Cause enumeration.

Table ‑. Damage Cause Enumeration

| Field Value | Description |
| --- | --- |
| 0 | Not Specified |
| 1 | Combat |
| 2 | Stochastic |
| 3 | Deterministic |

The following table provides the Damage Extent enumeration.

Table ‑. Damage Extent Enumeration

| Field Value | Description |
| --- | --- |
| 0 | Not Specified |
| 1 | Mobility Kill |
| 2 | Firepower Kill |
| 3 | Mobility and Firepower Kill |
| 4 | Catastrophic Kill |

The following table provides the Mission Type enumeration.

Table ‑. Mission Type Enumeration

| Field Value | Description |
| --- | --- |
| 0 | Not Specified |
| 1 | Indirect Fire |
| 2 | Close Air Support (CAS) |

The following table provides the Radar Track Status enumeration.

Table ‑. Radar Track Status Enumeration

| Field Value | Description |
| --- | --- |
| 0 | Not Specified |
| 1 | Radar Track Detected |
| 2 | Radar Track Broken |

### Comments

This section will be completed at a future release of this document.

# STEALTH

## Stealth State

Stealth state data shall be transmitted via the Application Control PDU (see section 5.2), with a Control Type of Status (7). The records typically used with Stealth State and Control are provided in Appendix A. The controlling application of the stealth may issue an Application Health Status Heartbeat Request (47240) record to indicate which application status information should be transmitted. The specific stealth application may define application-specific status values to represent various groupings of the stealth state / control records.

The stealth application shall follow the issuance rules of the Application State (47100) record (default heartbeat of 5 seconds). Any other records that are output on the state are application-specific to the stealth and those issuance rules are defined within the records.

The records defined for Stealth State / Control may be used with the Application Control PDU using the Data Query (2) and Data (3) Control Types.

## Stealth Control

Simulation control for the Stealth is performed via the Application Control PDU (see section 5.2), with a Control Type of Set Data (4). The records for usage with stealth control are further detailed in Appendix A. These records are currently only capable of supporting a single viewpoint. If the Stealth has multiple display capability, then it assumed that the additional displays will be controlled appropriately.

# NETWORK PROTOCOLS

## Network Transport Layer Rules

The VDIS Transport Level Protocol for DIS traffic is based on the IEEE 1278.2-1995, IEEE Standard for Distributed Interactive Simulation Communication Services and Profiles, herein referenced as the Communications Standards document. This standard establishes the requirements for the communication services to be used in a DIS application. This standard supports IEEE 1278.1-1995. The DIS Product Development Group (PDG) is currently in the process of revising IEEE 1278.2 and it’s one of SE Core’s goals to update it in accordance with the practices and policies set herein for network protocols.

The VDIS Communication Services definition employs the seven-layer Open Systems Interconnection Reference Model (OSIRM) defined in ISO 7498-1: 1994 and briefly described in the Communications Standards document. The standard identifies two profiles for use with existing DIS applications. VDIS will adopt the usage of Profile 2 – Multicast Service and the classes as identified in Table 4 of the standard (although note that the unicast protocol will be over IP, not IPMC). VDIS does not recommend the usage of Reliable Unicast (Class 3) for DIS PDUs because those PDUs requiring reliability have it built-in at a higher layer and the high overhead of TCP vs. UDP is not advisable.

VDIS recommends the usage of IPv6 as the new standard.

### Key Assumptions

Please refer to the assumptions about underlying requirements of the DIS application and how they will be applied in the Communications Standards document. The assumptions are as follows:

1. **Real-time considerations**: VDIS will define the timeliness required to support human-in-the-loop operation (using the Quality of Service parameter default values listed the Communications Standards document as the base).
2. **Long haul connection**: VDIS will support the connection of multiple sites to participate in a single exercise or a joint AAR by providing a common definition of the PDU protocol interactions and a common set of DIS enumerations to be used by VDIS-compliant training systems. Individual sites may have different network configurations to allow for optimization of their training, but gateways will need to convert to the long haul capability. Such gateways may employ the use of Reliable Unicast (e.g., TCP/IP) for lower level management, but that is outside the scope of this specification and is not required for VDIS interoperability.
3. **Multiple exercises**: VDIS will support the execution of multiple concurrent exercises at a single site. The method of assignment of an exercise ID is provided in Initialization and Asset Management. Different multicast groups or virtual networks may be utilized to isolate PDU traffic from the various exercises. The maximum number of exercises supported in VDIS is 32.
4. **Non-PDU traffic**: The VDIS communication services will support several types of data transmission that may or may not be directly related to DIS applications, though any one service instance will typically support only one type (e.g., DIS, CGF traffic).
5. **Communication management requirements**: VDIS does not recommend the usage of any application layer network management protocols such as Simple Network Management Protocol (SNMP). Such a requirement would inhibit various COTS solutions and provide perhaps too much overhead unneeded for VDIS applications. Instead, the DIS protocol will be the application layer protocol and UDP / IP (MC) will be the transport and internet layer protocols.
6. **Security**: Security requirements and methods will be defined on a case-by-case basis, as required by the training systems.

### Network Packet Filtering Approaches

The VDIS specification allows for the use of various methods that take advantage of standard network hardware to reduce the number of packets that are received at each simulation node. Major advances in network communications have occurred since the DIS protocol was first designed and more are on the way. VDIS currently will allow sites to take advantage of these capabilities without losing backward compatibility with the DIS standard but cannot mandate them since they cannot fully encompass the Virtual Training Domain.

#### Multicast Services

See section 4.2.1.2.2 of the Communications Standards document for the list of services required to support VDIS multicast. VDIS interprets the scope of these services to read “simulation network” where just “network” is printed. This is a clarification in that the network does not include the long haul network because long haul networking may restrict some multicast groups. Also, to be VDIS-compliant the list of recommended multicast services from section 4.2.1.2.3 of the Communications Standards document shall be followed. As noted in section 4.2.1.2.1, “The minimal form of best effort multicast services to support DIS shall consist of simultaneously transmitting to a group consisting of all hosts on a LAN.”

VDIS recommends the adoption of the following multicast groups specified by PDU, given in the recommended IPv6 format. For consistency, the IPv4 format is also provided.

* Simulation Traffic: ff02::1:e000:7XX (224.0.7.XX) – All other PDUs
* Entity Traffic: ff02::1:e000:9XX (224.0.9.XX) – Entity State
* Radio Traffic: ff02::1:e000:c0XX (224.0.192.XX) – Transmitter, Receiver, Signal

The “XX” for each group shall be replaced by the Exercise ID (zero-leading for IPv6). Thus, the Entity State group for exercise 1 is ff02::1:e000:901. VDIS does not restrict or impose a particular port. VDIS also does not specify other types of networks such as a system-level traffic, video traffic, or high-fidelity communications traffic; these are all TBD in this release.

#### Virtual Networks

High performance LAN switches can segment networks into logically defined virtual work group segments referred to as Virtual LANs (VLANs). This allows for the segmentation of switch ports to allow the communication between certain devices. This capability has been used to support the segmentation of real time simulation PDU data from other network traffic and to also separate data from different DIS exercises.

VDIS recommends, but does not require, the usage of VLANs. It’s important, however, that the simulation and Simulation Managers be aware of the existence and configuration of any VLANs so that they do not assign simulation applications to unavailable exercises.

### PDU Bundling

VDIS shall allow packet bundling (concatenation) of PDUs being sent onto the network. Because network routers, bridges, gateways, and computer hosts have a bound on the number of packets that they can process or transmit per second, a benefit can be derived from bundling smaller PDUs into larger PDU packets. The PDU packet bundling of multiple PDUs into one larger packet increases the peak transmission rate of these PDUs. Receiving simulation applications are required to support bundled PDUs.

The bundling of PDU traffic, when bundling is enabled, shall capture PDUs that share common routing attributes within the same frame and concatenate them into a single PDU buffer. When the number of bundled PDUs exceeds the maximum PDU buffer size, no additional PDUs are added to that buffer, the buffer is sent and a new buffer is started. When the time interval for a frame has expired, all PDUs for that frame are sent.

The common routing attributes that allow PDU bundling can include multicast PDUs sent on the same multicast groups or broadcast PDUs sent to the same destination (to a specific simulation application or to an entire exercise). PDUs that are issued for a local-send are not candidates for bundling because there will be no bandwidth savings and no need for the additional complication.

The ability to request the bundling of PDUs by type, size, destination, or content is not yet supported as an interoperable protocol. The reason for this would be fine tuning of network performance by one or more asset or simulation managers. VDIS does not yet provide a mechanism for this request by either a manager or a simulation application. Any such bundling parameters will be supplied in the scenario initialization file or configured manually on the simulation applications. This issue will be revisited later and thus is currently TBD.

There shall be no padding between PDUs that are bundled together. Normal padding that might be at the end of a PDU as defined in the standard is acceptable, but no additional padding is allowed. Bundling makes it critical that the PDU format is accurate and that the padding fields are required.

Some issuance rules require that certain PDUs be bundled with others. For example, a majority of the Attribute PDUs will need to be bundled with the PDUs that they modify or extend. If any such PDUs are required to be bundled, then the appropriate bit in the PDU Header PDU Status field shall be set.

## Performance Requirements

The performance requirements for VDIS refer to network bandwidth, quality of service (QoS), error detection, synchronization, and receipt and transmission of DIS PDUs. These performance requirements are the basis for selection of network hardware and software in a DIS exercise. See sections 4.2.2 and 5.2.2 of the Communications Standards document for discussions of the considerations for DIS performance requirements. VDIS recommends using Table 2 in the Communications Standards document for QoS requirements.

## Error Detection

VDIS-compliant products shall follow the Error Detection recommendations in section 4.2.3 of the Communications Standards document.

# TASK ORGANIZATION

The representation of task organization information in a training exercise provides exercise participants, exercise facilitators, and exercise managers insight into the simulation task organized units on the playing field. The information is especially useful to the Instructor Operator Station (IOS), the teams creating AAR analysis and briefing material, and the facilitators at the Role Player stations. The task organization information is usually introduced into a training exercise during scenario generation force laydown. The entities participating in the exercise can be task reorganized during exercise simulation.

A problem exists in distributing this task organization to the exercise. If the initial conditions scenario file contains a clear mapping between the simulated entities and the task organization structure, then any simulation application that has access to the file should be capable of presenting the task organization data to its users. The task organization information could be presented on the Plan View Display (PVD), on a menu display, or in the visual scene. For those simulation applications that do not have access to the scenario file, and for those situations where the task organization is changed during exercise simulation, this information can be distributed to exercise participants from the IOS station or from a role player workstation.

VDIS applications shall use the Entity Marking record with the U.S. Army Marking (2) character set. in the Entity State PDU to distribute task organization data, if such data is required for the simulation application. (NOTE—VDIS has updated the codes from SISO-REF-010 based upon information in the CCTT code baseline. See 4.6 for more information regarding these enumerations.) For those entities where task organization data is not available or necessary, then the Entity Marking record shall instead use ASCII (1) and provide a descriptive name.

The entities that do not require task organization data is left up to the individual training system to decide. However, OneSAF will support this updated Entity Marking character set and will be able to produce proper task organization data. The method for this is briefly identified here:

1. As the scenario is built, each echelon (unit) and entity is given an encoded string. This string can be any length (it is not directly related to a limit of 11 characters). The strength has a format similar to Individual/FT/Squ/Sec/Plt/Co/Bn/Div/Corps. The individual designation is something like LDR. The fire team through section may be omitted, as might the Division/Corps.
2. This string is stored in the ODM and converted in the VDIS interop manager to the encoded U.S. Army Marking character set.
3. Upon receipt of external entities, the encoded character set is converted to a similar string.
4. Optionally, OneSAF may provide a display-only string called a Bumper Number for the PVD or other types of reports finding shorthand useful. For example, instead of “1/1/A/3-32 Armor Reg” it could display “3-32AR A11.”
5. If the string cannot be converted into a known code, then instead the string shall be truncated to 11 characters and sent out as ASCII (1).
6. The last five octets of the Entity Marking record (bytes 8-12) are optional. If not encoded, they will be set to a blank (32).

## Aggregate State PDU for Task Organization

The use of the Aggregate State PDU protocol to distribute the task organization data was investigated. This effort would involve parsing the scenario initial conditions file for task organization data and creating the aggregate state PDUs to describe the organization structure. The Aggregate State PDU contains additional information not required for simply distributing the task organization data for each entity. It defines the task organization at each echelon in the hierarchical structure and allows attributes to be specified for each echelon node in the task organization. At each echelon level, an Aggregate State PDU would be used to define the parent organization (unit) and each of the children of the parent unit. The children could be both subordinate units and entities. While this effort would work, the amount of detail required to describe each aggregate unit appeared to be overkill for task organization definition. A big benefit to this scheme, however, is that it defines the task organization structure that shows the relationships between the entities and units in the task organization. For a training system that uses the Aggregate State PDU protocols to manage aggregation and de-aggregation for its PVD and visual scenes, the distribution of the task organization data would be derived at no additional distribution cost.

The Aggregate State PDU has the following category list for echelon level: Other, Individual Vehicle, Element, Platoon, Battery, Company, Battalion, Regiment, Brigade, and Division. The following echelon levels are not found in the Aggregate State PDU echelon levels: Squad, Team, Section, Squadron, and Troop. Use of the Aggregate State PDU to describe task organization data would require additional categories to add these missing echelon levels.

To use the Aggregate State PDU data to access the task organization for an entity, the simulation application must parse all of the Aggregate State PDUs to build the hierarchical structure that defines the relationships between all of the echelon levels and all of the subordinate units and entities contained within. To determine the task organization data for an entity from this aggregate state relationships data structure, visit the parent echelon for the entity to get its immediate echelon parent, and continue to traverse the parent chain for each echelon until the top of the echelon structure is reached. This list of parent nodes visited defines the task organization structure for the entity.

Because of the missing categories and inherent complexity in parsing the PDUs, the Aggregate State PDU is not suitable for Task Organization. It would be especially problematic for latecomers to the simulation exercise.

## Command from Simulator (CFS)

### Description

VDIS supports Command from Simulator (CFS) via the SIMAN family for action requests. This capability requires some coordination from the simulator (typically a manned module or its controlling application, or perhaps a live entity) and the task organization application (typically the SAF). Essentially, the scenario planning on the SAF system designates one or more *entities* as task reorganizable. These entities are then typically placed with a mixed-mode unit with actual CGF entities that cannot be replaced by manned modules or live trainers. This setup allows the exercise operator to embed scenario commands for the CGF entities that are expected to the follow the manned module.

### Initialization

Upon scenario creation, each entity that is expected to be run as a manned module (virtual trainer) is identified. These entities have an Entity Type and unique CFS Name. The Entity Type is the standard DIS enumeration, but the CFS Name is simply an ASCII string that is unique to the SAF system. The CFS Name recommended for VDIS is simply a concatenation of the task organization names in a bottom-up sequence, separated by slashes. For example, LDR/UnitName/CompanyB/BLUFOR defines the leader entity of a particular unit in a company on BLUFOR. The depth of the hierarchy is not limited by the VDIS protocol. VDIS also recommends that the comparison for task reorganization be case insensitive, despite the minor performance hit (if the CFS Name needs to be case sensitive for uniqueness there is likely an issue to be resolved). This format allows for easier transition on latecomers that are not privy to the scenario initial conditions file.

Entities that may become task organized shall set the Task Organizable bit (8) in the Capabilities record of the Entity State PDU. See section 4.7 for more information about this field.

### Task Reorganization

When the manned module comes on-line, it (or its simulation manager) sends an Action Request PDU with the Action ID Command from Simulator (48). This request contains the Command from Simulator (15505) variable datum record, as detailed in Appendix A. The SAF system that receives this action request shall respond within 10 seconds with an Action Response PDU containing a Request Status of Executing (2), Complete (4), or Request Rejected (5). If Executing (2) is received, then the manned simulator shall reset its timer and wait for additional Action Response PDUs. If Complete (4) is received, then the action as performed and the manned simulator shall consider itself task reorganized. The request shall fail with a Request Rejected (5) value when the CFS Name is not found (case insensitive), the SAF element has already been task reorganized, or the Entity ID is invalid. It may fail when the Entity Type does not match, although VDIS does not require that it fail (for example, an exercise agreement is possible that would allow an M1A1 to task reorganize an M1A2 because only an M1A1 manned module is available).

The SAF system should also reply to a Data Query PDU with the Command from Simulator (15505) datum ID. If received and the SAF system desires to handle these queries, then it shall reply with one or more Data PDUs, each one containing a Command from Simulator (15505) for one specific task reorganizable element (that hasn’t already been task reorganized and isn’t already in the process of being task reorganized). If there are no such entities, then the SAF system shall reply with an empty Data PDU. A Time Interval of 0 is the only valid value for the Data Query PDU with regards to this datum record.

# COMMUNICATIONS

## Transmitter PDU

The VDIS Transmitter PDU adheres to 1278.1-2012. VDIS provides the following clarifications:

1. VDIS neither recommends nor disallows usage of unattached radios. This is currently left as an exercise agreement.
2. If a radio system intentionally jams others (for audio or data), then the Input Source shall be set to Jammer (10).
3. Although VDIS doesn’t currently provide for specific VTP records, their usage is not forbidden.
4. The requirements on Modulations Parameters record are given in 9.3.1.

## Signal PDU

The VDIS Signal PDU adheres to 1278.1-2012.

## Tactical Data Link (TDL)

VDIS imposes the following additional requirements on usage:

1. Grouped digital messages in the same Signal PDU shall have the same Message Format and Version (see section 9.3.4.2).

### Modulation Parameters of Transmitter PDU

VDIS adopts the usage of the CCTT SINCGARS Modulation Parameters (MP) record. Additional issuance and receipt rules beyond DIS are also provided below.

#### Issuance rules

The following define additional specific field requirements:

1. FH Net ID. This field shall be set to zero when not in frequency hopping mode.
2. Hop Set ID. This field shall be set to zero when not in frequency hopping mode.
3. Lockout Set ID. This field shall be set to zero when not in frequency hopping mode.
4. FH Synchronization Time Offset. This field shall be set to zero when not in frequency hopping mode.
5. Transmission Security Key. This field shall be set to zero when not in frequency hopping mode.

#### Receipt rules

There are no additional receipt rules for this record.

### Link 16

VDIS adopts the usage of SISO-STD-002 (and PCR-001) for communicating Link 16 tactical messages (TDL J) within DIS. This is consistent with the CAF DMO standards.

### Link 11/11B

VDIS adopts the usage of SISO-STD-005 for communicating Link 11/11B tactical messages (TDL A/B) within DIS. This is consistent with the CAF DMO standards.

### EPLRS / SINCGARS / L-Band Satcom

VDIS strongly recommends inserting a minimum 1/10th of a second delay between the Transmitter PDU announcing the beginning of a digital transmission and the first Signal PDU containing the digital message. This is to accommodate the radio reception model within the manned module. There is no minimum required delay between the Signal PDU and the Transmitter PDU announcing the end of the transmission.

#### Transmitter PDU

VDIS requires that the Transmitter PDU fields be set according to the DIS standard, except for the following differences as appropriate for the digital messaging type. If the radio type is not known, then it shall default to EPLRS.

1. PDU Header. (No change from DIS standard.)
2. Entity ID. (No change from DIS standard.)
3. Radio ID. (No change from DIS standard.)
4. Radio Entity Type. The Category field shall be set as follows depending on the radio type. The nomenclature and nomenclature version fields are set accordingly, if applicable.
   1. EPLRS/SADL terminal (24)
   2. SINCGARS (27)
   3. L-Band Satcom (28)
5. Transmit State. (No change from DIS standard.)
6. Input Source. This shall be set to Digital Data Device (8).
7. Length of Variable Transmitter Parameters. (No change from DIS standard.)
8. Antenna Location. (No change from DIS standard.)
9. Relative Antenna Location. (No change from DIS standard.)
10. Antenna Pattern Type. (No change from DIS standard.)
11. Antenna Pattern Length. (No change from DIS standard.)
12. Frequency. The value for this field depends on the device, as follows:
    1. EPLRS – This value should be 435 MHz and shall be set to this if it is not within the range of 225 – 450 MHz.
    2. SINCGARS – This value should be 87.25 MHz and shall be set to this if it is not within the range of 30 – 87.975 MHz.
    3. L-Band Satcom – This value should be 1.616 GHz.
13. Transmit Frequency Bandwidth. The value for this field depends on the device, as follows:
    1. EPLRS – This value shall be 4 MHz
    2. SINCGARS – This value shall be 25 KHz
    3. L-Band Satcom – This value shall be 200 KHz
14. Power. (No change from DIS standard.)
15. Modulation Type. The Modulation Type fields contain enumerations for the major and detail modulation fields:
    1. Spread Spectrum. (No change from DIS standard.)
    2. Major Modulation. (No change from DIS standard.)
    3. Detail. (No change from DIS standard.)
    4. Radio System. This field shall be set as follows depending on the radio type:
       1. SINCGARS (5)
       2. EPLRS (7)
       3. L-Band Satcom (11)
16. Crypto System. (No change from DIS standard.)
17. Crypto Key ID. (No change from DIS standard.)
18. Length of Modulation Parameters. This field shall be set to 16 to correspond to the record as defined above in 9.3.1.
19. Modulation Parameters. These fields shall specify the modulation type specific characteristics of the Transmitter PDU.
    1. FH Net ID. This should be set to zero if not supported.
    2. Hop Set ID. This should be set to zero if not supported.
    3. Lockout Set ID. This should be set to zero if not supported.
    4. Start of Message. This should be set to Not start of message (0) if not supported.
    5. Clear Channel. This should be set to Not clear channel (0) if not supported.
    6. FH Synchronization Time Offset. This should be set to zero if not supported.
    7. Transmission Security Key. This should be set to zero if not supported.
20. Antenna Pattern. (No change from DIS standard.)
21. Variable Transmitter Parameters. (No change from DIS standard.)

#### Digital Message Data

Although more than one message is allowed per PDU for digital messaging, if multiple messages are passed then fragmentation is not allowed (otherwise it becomes problematic to parse the messages and smaller messages that are included up front could be unnecessarily delayed while the larger message gets built). These messages are potentially large (e.g., image data) and thus our digital messaging design supports fragmenting. Our model doesn’t require high fidelity implementation of the networking, but does need to support large messages and a certain amount of routing. So, we’ve designed a header that allows fragmentation and pulls out several key fields from the encoded message (typically VMF), because the encoding would be difficult to reproduce at the gateway or even higher in the application layer of the receiving simulation. Additionally, the process to do so without foreknowledge is trial and error and computationally expensive especially when multiple header types or message formats need to be accommodated because there is no one parser that knows everything. The following is the format for the digital message structure used within the Signal PDU to support EPLRS, SINCGARS, and L-Band Satcom.

1. **Reserved**. VDIS reserves 160-bits of data at the beginning of the message in order to support a future high-fidelity radio communications model. This also makes the format of the digital message consistent with the other Link standards.
2. **Header Length**. This 16-bit unsigned integer identifies the length in octets of this header, including this field. This length does not include the message data itself.
3. **Message Format**. This 8-bit enumeration identifies the format of the encoded message. The enumeration follows:
   1. 0-15 – Follows the enumeration for the 47001C User Message Format (UMF) field. Please refer to 47001C for a description of each.
      1. 0 – Link 16 (J-series) message
      2. 1 – Binary File
      3. 2 – Variable Message Format (VMF) (K-series) message
      4. 3 – National Imagery Transmission Format System (NITFS)
      5. 4 – Forwarded Message (FWD MSG)
      6. 5 – United States Message Text Format (USMTF)
      7. 6 – DOI-103
      8. 7 – eXtensible Markup Language (XML) – Message Text Format (MTF)
      9. 8 – eXtensible Markup Language (XML) – Variable Message Format (VMF)
      10. 9-15 – Undefined
   2. 16 – AFAPD
4. **Version**. This 8-bit enumeration identifies the version and is specific to the Message Format (there will be a different Version enumeration for each format).

The enumeration for a Message Format of VMF (2) follows:

* 1. 0 – TIDP-TE R2
  2. 1 – TIDP-TE R3
  3. 2 – TIDP-TE R4
  4. 3 – TIDP-TE R5
  5. 4 – TIDP-TE R6
  6. 5 – TIDP-TE R7
  7. 6 – 6017
  8. 7 – 6017A
  9. 8 – 6017B
  10. 9 – 6017C
  11. 10-15 – Reserved
  12. 16 – DCX2\_AV

The enumeration for a Message Format of USMTF (5) follows:

* 1. 0 – 1993
  2. 1 – 1995
  3. 2 – 1997
  4. 3 – 1998
  5. 4 – 1999
  6. 5 – 2000
  7. 6 – 2001
  8. 7 – 2002
  9. 8 – 2003
  10. 9 – 2004
  11. 10-15 – Reserved

The enumeration for a Message Format of AFAPD (16) follows:

* 1. 0 – AFAPD AV
  2. 1 – AFAPD BG

1. **Message ID**. This 32-bit unsigned integer defines the numerical, increasing ID of the message and is typically used for identifying parts of fragmented messages. See the section on fragmentation rules for more information. This number does not generally refer to a value within the encoded digital message. If the Message ID increments beyond the maximum value for a 32-bit integer, then it shall wrap around to 0. The Message ID should start at 0.
2. **Segment Number**. This 32-bit unsigned integer defines the segment number contained in this PDU for the current, fragmented message. If only one message is required, then this field shall be zero (not one of one). See the section on fragmentation rules for more information.
3. **Total Segments**. This 32-bit unsigned integer defines the total number of segments in a fragmented message. If only one message is required, then this field shall be one. See the section on fragmentation rules for more information.
4. **Message Number**. This 32-bit record identifies the specific message. The definition of this record depends on the Message Format as identified below. In the cases where the acknowledgment tactical message doesn’t have the FAD/Message Number (for UMF) or Message Type/Subtype (for AFAPD), then the simulation VDIS acknowledgment message for this record (Message Number) shall use the values from the message being acknowledged.
   1. *47001C UMF*. This record is defined as follows:
      1. Functional Area Designator (FAD) – 16-bit unsigned integer
      2. Message Number – 16-bit unsigned integer
   2. *AFAPD*. This record is defined as follows:
      1. *Message Type* – 16-bit unsigned integer
      2. *Message Subtype* – 16-bit unsigned integer
5. **Source Address**. This 64-bit string identifies the source of the message. This is an alphanumeric string that will be null-terminated only if less than 8 characters long.
6. **Number of Info Messages**. This 8-bit unsigned integer indicates the number of informational messages contained in the PDU. The classification of an info message is based on the VMF structure.
7. **Precedence**. This 8-bit enumeration identifies the precedence of the message and matches 47001C as follows. If the Message Format doesn’t support precedence, then this field should be set to Undefined (0) or Routine (7).
   1. 0-1 – Undefined
   2. 2 – Emergency
   3. 3 – Undefined
   4. 4 – Flash
   5. 5 – Immediate
   6. 6 – Priority
   7. 7 – Routine
8. **Classification**. This 8-bit enumeration identifies the classification level of the message as follows:
   1. 0 – Unclassified
   2. 1 – Confidential
   3. 2 – Secret
   4. 3 – Top Secret
9. **Destination Type**. This 8-bit enumeration identifies the type of destination addresses used. All destinations shall have the same destination type. This enumeration is defined as follows:
   1. 0 – URN. This is stored as a character string of up to 8 characters representing a decimal integer value. e.g., 1,509,713 would be stored as a null-terminated string “1509713”.
   2. 1 – IPv4. This is stored as 8 hexadecimal characters, not null-terminated. For example, 225.0.1.95 would be stored as “E100015F” (upper or lower case may be used).
   3. 2 – Subscriber ID. This is stored as an 8-bit unsigned integer followed by 56-bits of padding.
10. **Number of Destinations (N)**. This 8-bit unsigned integer defines the number of destination addresses stored in this header. There is no requirement that this field be 1 or higher. If this field is zero then no destination addresses follow.
11. **Destination Address**. These 64-bit records identify each of the destinations for this message. There shall be 0 to 16 possible destinations. The record definition depends upon the Destination Type.
    1. 0-1 – 8-character string, null-terminated if less than 8-characters long. For IPV4 (1), it is guaranteed to be 8-characters.
    2. 2 – Subscriber ID. This record is defined as follows:
       1. *ID* – 8-bit unsigned integer
       2. *Padding* – 8-bits unused
       3. *Padding* – 16-bits unused
       4. *Padding* – 32-bits unused

Table 9‑1. VDIS Digital Message

| Field | Type | Range |
| --- | --- | --- |
| EPLRS, SINCGARS, and L-BAND Satcom Digital Message Header | | |
| Reserved | 160-bits reserved | N/A |
| Header Length | 16-bit unsigned integer | N/A |
| Message Format | 8-bit enumeration | See above |
| Version | 8-bit enumeration | See above |
| Message ID | 32-bit unsigned integer | N/A |
| Segment Number | 32-bit unsigned integer | N/A |
| Total Segments | 32-bit unsigned integer | N/A |
| Message Number | 32-bit record | N/A |
| Source Address | 64-bit string (8characters) | N/A |
| Number of Info Messages | 8-bit unsigned integer | 0-16 |
| Padding | 8-bits unused | N/A |
| Padding | 16-bits unused | N/A |
| Precedence | 8-bit enumeration | See above |
| Classification | 8-bit enumeration | See above |
| Destination Type | 8-bit enumeration | See above |
| Number of Destinations (N) | 8-bit unsigned integer | 0..16 |
| Destination Address 1 | 64-bit record | N/A |
| … | … |  |
| Destination Address N | 64-bit record | N/A |
| EPLRS, SINCGARS, and L-BAND Satcom Digital Message Data | | |
| Message Data | varies | N/A |

#### Signal PDU

VDIS requires that the Signal PDU fields be set to the following values as appropriate for the digital messaging type.

1. PDU Header. (No change from DIS standard.)
2. Entity ID. (No change from DIS standard.)
3. Radio ID. (No change from DIS standard.)
4. Encoding Scheme.
   1. The Encoding Class (bits 14-15) shall be set to Raw Binary Data (1).
   2. The Number of TDL Messages (bits 0-13) shall contain the number of digital messages contained within this PDU. Please refer to the section on fragmentation rules for more information.
5. TDL Type. This field shall be set accordingly as follows:
   1. 22 – EPLRS
   2. 24 – SINCGARS
   3. 106 – L-Band Satcom
6. Sample Rate. This field shall be set to a value appropriate to the current signal message and device. For example, this could be 4,800 for VMF messages on the Fire Support Net and 16,000 for VMF messages on the Tactical Internet.
7. Data Length. (No change from DIS standard.)
8. Samples. This field shall be set to zero.
9. Data. This section of the PDU corresponds with the digital messaging data as defined in section 9.3.4.2.

The Signal PDU for use in EPLRS, SINCGARS, and L-Band Satcom TDL messages is illustrated below.

Table ‑. Signal PDU with VDIS Digital Messages

| Field(s) | Value(s) |
| --- | --- |
| PDU Header | See IEEE 1278.1 |
| Entity ID | See IEEE 1278.1 |
| Radio ID | See IEEE 1278.1 |
| Encoding Scheme (bits 14-15) | Raw Binary Data (1) |
| Encoding Scheme (bits 0-13) | See above |
| TDL Type | EPLRS (22), SINCGARS (24), or L-Band Satcom (106) |
| Sample Rate | 0 |
| Data Length | See IEEE 1278.1 |
| Samples | 0 |
| Digital Message Data | See section 9.3.4.2 |

#### Fragmentation Rules

The following define the rules regarding fragmented messages.

1. If a message is fragmented, then it shall be the only message contained within a particular Signal PDU.
2. The Number of TDL Messages shall be set to 1 when containing fragmented (parts of) messages.
3. The sender shall increment the Message ID whenever a new digital message is sent.
4. The sender shall increment the Message ID whenever a digital message is retransmitted, such as due to a lack of an acknowledgment.

# LOGISTICS

VDIS adopts the Logistics usage defined in DIS, except for the following sections.

## Repair

* VDIS recommends different values for the following symbolic names.
  + Repair Receiver Timer 1 (REPAR\_REC\_T1): 1 minute (1278.1-2012 default 5 seconds)
  + Repair Supplier Timer 1 (REPAR\_SUP\_T1): 130 seconds (1278.1-2012 default 12 seconds)
  + Repair Supplier Timer 2 (REPAR\_SUP\_T2): (unchanged) (1278.1-2012 default 12 seconds)
* Upon receiving a Service Request PDU to which it desires to respond in the affirmative, the repairing entity shall transmit a Data Query PDU to request the information about the repair operation. The datum ID supporting for the query is Component Damage (35400) and the variable record structure for this is provided in Appendix A.
* The receiving entity shall respond to the Data Query PDU with all of the repairs that it wishes performed.
* The receiving entity shall timeout its repair request after 5 minutes if no Data Query PDU is received. This timeout constitutes a condition for the repair service no longer being met.
* If the repairing entity wishes to deny the repair request, it shall transmit a Repair Complete PDU with the Repair field set to No Repairs Performed (0). This rejection constitutes a condition for the repair service no longer being met.
* Upon completion of a repair, the repairing entity shall identify either All Requested Repairs Performed (1) in the Repair field of the Repair Complete PDU or the specific system repaired. The receiving entity shall proceed into the Ready state unless not all repairs are performed, then it shall reevaluate its need for further repairs or even recovery.

## Recovery

If repairs cannot be performed then a recovery operation may occur. Additionally, a recovery operation could be determined by the receiving entity (or operator) or the repairing entity (or operator). The protocol for recovery shall be as defined for Towing in section 4.14.2.

## Resupply

VDIS recommends these values for the following symbolic names:

* Resupply Receive Timer 1 (RESUP\_REC\_T1): (unchanged) (1278.1-2012 default 5 seconds).
* Resupply Receive Timer 2 (RESUP\_REC\_T2): (see below) (1278.1-2012 default 55 seconds).
* Resupply Supplier Timer 1 (RESUP\_SUP\_T1): (see below) (1278.1-2012 default 1 minute).

When the minimum time to transfer a single unit of a supply item is greater than 55 seconds, then the exercise agreement shall be modified to increase the timers (“Resupply Receive Timer 2” and “Resupply Supplier Timer 1”) accordingly. VDIS does not currently provide a table of modified timers for specific munitions or other supplies.

Additionally, VDIS appends the logistics protocols regarding resupply as identified in DIS as follows:

* A resupply offer with a quantity of zero is effectively treated as a rejection, but the receiving entity shall still reply with a resupply received quantity of zero. No Resupply Cancel PDU is required. A quantity of zero for one item does not indicate a complete rejection if there are other items that have a non-zero quantity, in which case the resupply would still take place without having to reinitiate it.
  + All liquids used in resupply shall use units of measure in liters (not cubic meters), unless explicitly specified otherwise. Thus, the following supply types shall be in liters (asterisk denotes all sub-elements):
* 6.0.0.0.1.\*.\* – Fuels
* 6.0.0.0.2.\*.\* – Oils
* 6.0.0.0.4.1.\* – Water drinking sterile clear colorless
* 6.0.0.0.4.2.\* – Water drinking distilled or sparkling

## Future Capabilities

In the future, VDIS may consider extending the Resupply protocol in the following areas:

* As an alternative to a modified exercise agreement concerning the resupply timers identified above, the Service Request PDU could be modified to include a minimum time in seconds of the longest single item to supply. This time would then effectively become the per-instance value of Resupply Receive Timer 2, and Resupply Supplier Timer 1 would simply be 5 seconds greater.
* As an alternative to continual hand-shaking for many items (of even the same type), the Resupply Received PDU could be modified to contain a state for continued resupply service.

# DYNAMIC ENVIRONMENT

## Dynamic Environment Objects

The VDIS recommendation for the dynamic environment objects listed below is consistent with the object state PDUs described in DIS, except as noted below. The VDIS object origin approach was adopted in IEEE Std 1278.1-2012. The Environment Manager for usage with the DIS object approach to synthetic environments will be described elsewhere.

It is often not clear as to whether a particular battlespace object should be an entity (processed via the Entity Information/Interaction family) or object (processed via the Synthetic Environment family). The following criteria can help with these decisions. The criteria below are marked as “strong” or “weak” to help resolve conflicts on difficult cases.

To characterize a feature as an object it should meet the following criteria:

1. Strong: It can be emplaced as part of the terrain (as a terrain feature).
2. Strong: It is generally stationary feature (does not move on its own; is not portable or easily portable).
3. Strong: It has no attached parts or articulations.
4. Weak: Its modeling is simple (state changes limited to damage and breach states).
5. Weak: Items that have close associated with Synthetic Natural Environment (SNE) services (e.g., they alter height terrain) should be given consideration of objects as well, to localize the effects.

To characterize a feature as an entity, it should meet the following criteria:

1. Strong: It is often (or possibly) self-mobile or moved.
2. Strong: It is strongly associated with moving objects, vehicles, etc. (e.g., supplies).
3. Strong: It has attached parts or articulations.
4. Weak: The item initiates interactions with other entities.
5. Weak: It has complex modeling (triggers, behaviors).
6. Weak: It can, and often is, transported (carried) by another entity.

This rationale would conclude that features like trash bags, trash cans, and cardboard boxes that could be used as an Improvised Explosive Device (IED) would be characterized as entities instead of objects because they may be rigged with explosives to become IEDs and that they can (at least conceptually) be easily be relocated on the playing field.

Armored Vehicle Launched Bridges (AVLBs) are a close call. They could be characterized as entities because they are transportable. However, AVLBs have strong effects on common SNE services like elevation and terrain type. Therefore, although the vehicle itself is an entity and the bridge mechanism is an articulated and attached part; once deployed, the bridge itself becomes an object.

Table 11‑1 presents an exemplary list of battlespace objects and how they would be classified according to the above criteria. Any feature that is on the list as an object is not recommended to be transmitted or otherwise supported as an entity. For example, even though there are building entities, this table recommends only using object states for buildings. Using this delineation will maximize interoperability because environment services will function correctly.

Table ‑. Terrain Feature Characterization

| Terrain Feature | Characterization |
| --- | --- |
| Log Crib Rectangular | Point Object |
| Dragons Teeth | Point Object |
| Abatis 8 Tree | Point Object |
| Boresight Target Panel | Point Object |
| Zeroing Target Panel | Point Object |
| Rock Drop Covered | Point Object |
| Rock Drop Uncovered | Point Object |
| Crater Small | Point Object |
| Crater Medium | Point Object |
| Crater Large | Point Object |
| AVLB M60A1 | Point Object |
| AVLB MTU20 | Point Object |
| Ribbon Bridge | Point Object |
| NBC Hazard Marker | Point Object |
| Armored Vehicle Defilade | Point Object |
| Fighting Vehicle Defilade | Point Object |
| Mortar Carrier Defilade | Point Object |
| Tank Defilade | Point Object |
| Infantry Fighting Position (Non-Covered) | Point Object |
| Covered Machine Gun Bunker | Point Object |
| Machine Gun Prepared Position (Non-Covered) | Point Object |
| Overhead Covered Infantry Position | Point Object |
| Insurgent Building | Point Object |
| Tank Ditch | Linear Object |
| Concertina Wire | Linear Object |
| Concrete Barrier | Linear Object |
| Minefield Lane Marker | Linear Object |
| Breach | Linear Object |
| Minefield, Hasty | Areal Object |
| Minefield, Prepared | Areal Object |
| Minefield, Scattered | Areal Object |
| Minefield, Solitary | Areal Object |
| Trash Cans, Small | Entity |
| Trash Cans, Medium | Entity |
| Trash Cans, Large | Entity |
| Trash Bags, Small | Entity |
| Trash Bags, Medium | Entity |
| Trash Bags, Large | Entity |
| Trash Bags, Extra Large | Entity |
| Cardboard Boxes, Small | Entity |
| Cardboard Boxes, Medium | Entity |
| Cardboard Boxes, Large | Entity |
| Boxes, Small | Entity |
| Boxes, Medium | Entity |
| Boxes, Large | Entity |

No changes for the Object General Appearance from the documented bitfield enumerations.

### Point Object State PDU

No changes from the documented bitfield enumerations.

### Linear Object State PDU

The VDIS Linear Object State PDU changes were adopted in IEEE Std 1278.1-2012, specifically regarding the misalignment of the appearance fields. IEEE Std 1278.1-2012 has altered this PDU by changing all of the segment dimension parameters to 32-bit floats. This means that Segment Length, Width, Height, and Depth in the Linear Segment Parameter record (and consequently the Linear Object State PDU) will all be 32-bit floats, and thus the record and PDU as a whole will be 64-bits larger. Although VDIS does not currently recommend this change due to impacts on confederates, consideration should be taken to initiate adoption.

No changes for Appearance from the documented bitfield enumerations.

### Areal Object State PDU

No changes from the documented bitfield enumerations.

## Weather

The VDIS recommendation for Weather is consistent with the Environmental Process PDU described in DIS, with the corresponding state and geometry records defined in Appendix A.

To communicate atmospheric environmental parameters within an exercise, VDIS recommends using the Environmental Process PDU to distribute weather changes. The Environmental Process PDU is specifically defined to communicate simple environment variables and small scale environmental updates within an exercise. The Environmental Process PDU provides the capability to communicate the weather state and the region covered by that weather state on the playing field.

### Weather Manager

SE Core usage of the environmental process PDU allows distribution of the current weather state to exercise participants. VDIS recommends usage of a Weather Manager as shown in the diagram in Figure 11‑1.



Figure ‑. Weather Manager

When no dynamic weather conditions are being distributed within the exercise, the Weather Manager operates in ‘quiescent’ mode. The ‘quiescent’ mode does not require periodic heartbeat for weather conditions. Only weather changes are announced in the exercise. The client simulation applications do not need to be made aware of whether the Weather Manager is operating in quiescent mode or not, they receive the weather updates based on the output rate determined by the Weather Manager, The Weather Manager must provide the capability to distribute current weather conditions to simulation applications that join the exercise as a late comer. The Environmental Process PDU provides the capability to extend weather modeling to localized (regional) weather effects.

### Global and Regional Weather Identification

Global and regional weather data shall not be included in the same PDU. Multiple regions should be defined in different Environmental Process PDUs because they are different processes. Once multiple regions are defined in the same PDU, they can never again be separated because they will be associated with the same ID. Global weather data from the source simulation application shall use the same Environmental Process ID for the duration of an exercise. A global weather environmental process shall never be terminated (the Environment Status field shall be set to “2” which indicates that the second bit is On). A specific weather region shall also use the same Environmental Process ID for the duration of an exercise (though a different ID than the global weather or other regions, and this ID may actually be for multiple, linked regions). If a region is terminated, then a final Environmental Process PDU with the Environment Status field set to “1” shall be transmitted.

### Weather Records

A set of weather state and geometry records is provided in Appendix A to communicate the set of weather conditions. Multiple regional weather state records can be defined to provide multiple regions with multiple layers within a region, as assigned to different geometry records according to the usage of the Environmental Process PDU.

A Geometry Record is defined within the Environment Process PDU environment record to specify the extent of coverage for the weather state. The Uniform Geometry Record is used to specify that the weather attributes within a weather record are to be applied globally across the playing field. We recommend usage of the Rectangular Volume Record 2 Geometry Record to specify a region to be attributed to a set of regional weather state attributes. The regional weather state records are used to specify weather regions as well as to define layers within a region. The Weather Manager application will determine the resultant weather conditions when one or more regional weather state record attributes are applied in concert with the global weather state record attributes during exercise simulation. VDIS provides a more detailed context of the Uniform Geometry, Rectangular Volume Record 2, and Bounding Sphere geometry records within the Environmental Process PDU.

The ordering of geometry and state records within the Environmental Process PDU is important. A geometry record shall precede the state record(s) that apply to that volume. More than one state record may follow a geometry record, but there must always be at least one state record (though none follow the bounding sphere should one exist). If there are multiple volumes (regions) within a single PDU (indicating that these regions are forever linked with a single process ID), then a Bounding Sphere record shall be included as the first record in the PDU and shall enclose all volumes described within the PDU.

Below are several examples of records within the context of VDIS. The first example illustrates how global weather data should be transmitted. Note that there is no bounding sphere and only one geometry record, the Uniform Geometry record. The second example shows a single region. Note again that there is no bounding sphere. The third example shows multiple regions. Note now that a bounding sphere is included. Note also how each example uses a different set of state records. There is no ordering requirement for the state records or any requirement for specific state records to be included.

Table ‑. Example of Global Weather

|  |
| --- |
| Uniform Geometry |
| Weather State Atmospheric |
| Weather State Celestial |
| Weather State Wind |
| Weather State Precipitation |

Table ‑. Example of Regional Weather

|  |
| --- |
| Rectangular Volume Record 2 |
| Weather State Atmospheric |
| Weather State Wind |
| Weather State Precipitation |

Table ‑. Example of Multiple Weather Regions

|  |
| --- |
| Bounding Sphere |
| Rectangular Volume Record 2 |
| Weather State Celestial |
| Weather State Atmospheric |
| Weather State Wind |
| Rectangular Volume Record 2 |
| Weather State Precipitation |

### Weather Environment Type

VDIS does not recommend using specific Environment Types when transmitting global or regional weather data. The simulation application transmitting the Environmental Process PDU shall use zeros for the individual fields of the Environment Type. Receiving simulation applications shall not process this field or shall accept any values in this field to account for a forward-looking approach (e.g., a potential value for Kind could be “4” which is an Environmental entity).

## Minefields

### Areal Object State

VDIS recommends the usage of Areal Object State PDUs to support minefield interactions. These PDUs are already in place for legacy systems, so no effort is planned to step-up to the Minefield Family of PDUs (which would require extension to support legacy compatibility). The 5 objects (1 linear, 4 areal) to be used for minefields are listed above in Table 11‑1. The minefields themselves shall be represented by areal objects, designated as hasty, prepared, scattered, or solitary. The exact meanings of these types of minefields are not within the scope of this specification. Minefield-related event reports must be captured as documented in section 5.5.10.

### Minefield Breaching

The linear object, Breach, shall be used to create breach lanes in the minefield for which breach lanes are known to exist. The Breach field (bits 0-1) of the Areal Object State Specific Appearance for Minefields (11.3.1) shall be set to Breached (1). Minefield Lane Marker Breach may optionally be used to mark the left, right or both sides of the minefield breach. How and why these breach lanes are created by models, how they are modeled in the SAF, and whether they are visualized are areas beyond the scope of this specification.

## Improvised Explosive Devices (IEDs)

The requirements for improvised explosive devices are defined in this section. IEDs fall into two primary categories: munitions and non-munitions. The former is explicitly an identified munition, such as a 155-mm shell. This categorization is not only common in the real world, but also common for CGF implementations and usage in virtual training systems. Munitions are convenient because little additional work is required for them. For example, if a simulation already supports vulnerability models for the 155-mm artillery round, then adopting that munition as an IED is nearly a free capability.

IEDs that are not explicitly munitions (even jury-rigged munitions) could be any of the following, as examples: flammable liquid, fertilizer, chemicals, TNT belt, or biological weapons. Some or all of these types of devices may be useful, if not directly necessary, to support the individual training exercise requirements. For instance, the IED used in the 1995 Oklahoma City bombing was a rental truck containing 6,200 pounds of ammonium nitrate fertilizer, nitromethane, and a diesel fuel mixture (a total equivalent of over 2,300 kg of TNT).[[1]](#footnote-1)

IEDs shall be one of three types: detonation-based, entity-based, and object-based. The following requirements pertain to all types or by each type:

1. The detonation of an IED shall use the Munition or Explosion Descriptor record.
2. **Detonation-based.** These are “magic” in the sense that the IED itself is not conveyed as an entity or object in the battlespace. The first and only indication of the IED is via a Detonation PDU (no prior Fire PDU).
   1. The Source Entity ID in the Detonation PDU shall be NO\_SPECIFIC\_ENTITY if the detonation is caused by a munition.
   2. The Exploding Entity ID in the Detonation PDU shall be NO\_SPECIFIC\_ENTITY.
   3. The Event ID in the Detonation PDU should be set to NO\_EVENT\_ID.
3. **Entity-based.** These IEDs are present on an existing entity or are the entity itself.
   1. The IED Presence Indicator in the Entity State PDU Capabilities field shall be used to indicate the presence of an IED.
   2. An Attached Part VP record may be used to identify a specific IED and how it’s attached to the main entity. An Attached Part record is not always needed, however, such as with a munition that is itself an IED.
   3. The Extended Platform Appearance VP record may be used to identify an IED on a platform.
   4. The Extended Life Form Appearance VP record may be used to identify an IED on a life form.
   5. An Articulated Part VP record may be used to identify a specific IED and how it’s attached to the main entity.
4. **Object-based.** These IEDs are present on an existing object, but are not the object itself.
   1. The Object State General Appearance shall be used to indicate the presence of an IED.
   2. An Attribute PDU with an Object/Entity Association record should be used to associate an entity type with the object.

### Object State General Appearance

The General Appearance record for all Object State PDUs includes an IED Present field. This field should be used to indicate the presence and visual detection of IEDs in the object.

### Object/Entity Association record

To provide a specific IED on the object would require something similar to the Attached Parts VP record. However, there are no VP records in the Object State PDUs, so the only alternative is to use an Attribute PDU. The Object/Entity Association record is an Attribute record that supports this.

#### Purpose

Associations between objects and entities shall be communicated with the Object/Entity Association record in an Attribute PDU coupled with the Object State PDU for the object. This record correlates to the Entity Association VP record in the Entity State PDU for the corresponding entity, if that entity exists in the battlespace.

#### Record Definition

The following fields define the Object/Entity Association record:

1. **Record Type**. This 32-bit enumeration identifies the type of Attribute record.
2. **Record Length**. This 16-bit unsigned integer specifies the total length in octets of the Attribute record, including the type and length itself.
3. **Entity Type**. This 64-bit record identifies the entity type that is associated with this object.
4. **Entity ID**. This 48-bit record identifies the specific entity that is associated with this object, if known.
5. **Own Station Location**. This 16-bit enumeration indicates the station location on the object itself where the entity is physically connected, if known. The station location shall be as defined in SISO-REF-010. If there is more than one physical connection, then each connection that is described shall be included in a separate Object/Entity Association record.

Table ‑. Object/Entity Association Record

| Field | Type |
| --- | --- |
| Attribute Record Type | 32-bit enumeration |
| Attribute Record Length | 16-bit unsigned integer |
| Padding | 16 bits unused |
| Entity Type | 64-bit Entity Type |
| Entity ID | 48-bit Entity Identifier |
| Own Station Location | 16-bit enumeration |
| Padding | 64 bits unused |

#### Issuance Rules

The following issuance rules apply:

1. This record shall be transmitted in an Attribute PDU coupled with an Object State PDU (the specific type depending on the object type).
2. This record may be used to associate an object with an entity type, even if an appropriate entity doesn’t exist in the battlespace. An example of this is to assign an entity type munition as an IED on the object.

The following specific field requirements apply:

1. If the associated entity type does not have an entity ID, then NO\_ENTITY shall be used.

#### Receipt Rules

There are no specific receipt rules for this record.

## Tactical Smoke

The current VDIS recommendation for tactical smoke is to model it locally based upon receipt of Entity State PDU information (smoking, trailing dust, etc.), Fire PDUs, and Detonations PDUs and not broadcast/multicast updates over the simulation network. A simulation may use existing DIS methods (such as Object State PDUs) to propagate tactical smoke data within its local real-time simulation network, but other applications on that network that model the tactical smoke locally should ignore those methods.

## Breaching Point and Linear Objects

Refer to section 11.3.2 on minefields for breaching of Areal objects. VDIS recommends using the General Object Appearance record’s Damage field (bits 8-9) and Specific Appearance record’s breach fields (e.g., Breach State) of the Object State PDU for point and linear object breach visualization. The two appearance flags, Is Smoking and Is Flaming, are not used to indicate breached status, but may be set as appropriate under all breached conditions.

When no breaching is present, the Breach State field in the Specific Appearance record (if available for that object type) shall be set to No Breaching (0) and the Damage flag may be set as appropriate. When a point or linear object is breached, but not cleared, the Damage flag shall indicate Damaged (1). When a point object is breached, but not cleared, the Breach State flag shall indicate Breached (1). When a linear object is breached, but not cleared, its Breach State flag shall indicate Slight Breaching (1) or Moderate Breaching (2) as appropriate. When a point or linear object is breached and cleared, the Damage flag shall indicate Destroyed (2). When a point object is breached and cleared, the Breach State flag shall indicate Cleared (2). When a linear object is breached and cleared, its Breach State flag shall indicate Cleared (3). Even if an object’s Breach State flag is Cleared (2 or 3), its State shall not be set to Deactivated (1) unless the simulation system desires not to simulate the object anymore (and this determination might depend upon an exercise agreement). The State transition is not necessarily related to the breaching condition of the object.

## Dynamic Terrain

To support dynamic terrain, VDIS adopts the usage of two process-based PDUs: Scaled Height Map PDU and Soil Attribute Map PDU.

1. Scaled Height Map PDU. Communicates heights at regular intervals inside a predefined rectangular area.
2. Soil Attribute Map PDU. Communicates soil attributes inside a predefined rectangular area of the terrain.

### Scaled Height Map PDU

#### Purpose

The Scaled Height Map PDU communicates heights at regular intervals inside a predefined rectangular area. Each data item in the PDU represents a height post along the grid. The heights are scaled in order to allow more of them to be packed in a PDU. This implementation is able to send twice as many points in one PDU as a non-scaled implementation.

#### Scaling Precision

Scaling the points, however, may mean that some precision is lost in situations where the elevation offset of a post is not a multiple of the scaling value. Although loss of precision is not usually acceptable, the amount of precision lost is not problematic since it ranges in the 5th (hundred-thousandth) digit of precision. For example, if the range of values to be scaled is 10 units, then the values will never contain an error greater than 0.0000762951 units (half of the scale value).

#### Information contained in the Scaled Height Map PDU

The Scaled Height Map PDU shall contain the following information:

1. Standard PDU header data
2. Identification information for this terrain update
3. Data describing the area of terrain being updated
4. Data describing the height of terrain throughout the area being updated
5. The scale at which the data is being reported

#### Issuance of the Scaled Height Map PDU

The Scaled Height Map PDU shall be issued by a simulation application in response to a simulation event or occurrence that alters the terrain skin. If the terrain area that has changed is large, multiple PDUs will be sent. The PDUs will be associated via matching Object ID and Update Number fields, and the PDU Number and PDU Total fields.

The data within the PDU is calculated via the following high-level algorithms:

* Define the area to be gridded. The area must be a level plane.
  + Origin GCC (for the sake of clarity, consider this the Bottom Left of the area)
  + Vector to Top Left Corner = (Top Left Corner GCC - Origin GCC)
  + Vector to Bottom Right Corner = (Bottom Right Corner GCC - Origin GCC)
  + Unit Vector to Top Left Corner = normalize(Vector to Top Left Corner)
  + Unit Vector to Bottom Right Corner = normalize(Vector to Bottom Right Corner)
* PlaneUnitNormal = NORMALIZE (CROSS (Vector to Bottom Right Corner, Vector to Top Left Corner))
* Let R = row
* Let C = column
* RowStep = NORMALIZE (Vector to Top Left Corner) \* length (Vector to Bottom Right Corner) / R
* ColumnStep = NORMALIZE (Vector to Bottom Right Corner) \* length (Vector to Top Left Corner) /C
* For each C and R
  + ElevationTestGCC = Origin GCC + (Unit Vector to Top Left Corner)\* (R \* RowStep) + (Unit Vector to Bottom Right Corner)\*((C \* ColumnStep)
  + PlaneElevation[C,R] = ElevationTestGCC.convertToGDC().elevation
  + TerrainElevation[C,R] = Query the terrain elevation at the location created by following the PlaneUnitNormal from the ElevationTestGCC until it intersects the terrain.
* Range = MAXIMUM (TerrainElevation) – MINIMUM (TerrainElevation)
* Scale = Range / 65535 ***NOTE: Using the value ‘65535’ allows us to normalize the elevation offsets to the range of a 16-bit data type*.**
* PduHeight[C,R] = (TerrainElevation[C,R] – PlaneElevation[C,R]) / Scale

The following helps to describe the area in further detail:

* The plane of the rectangular area is coplanar with the lowest point in the grid.
* The two vectors should be orthogonal (Vector to Top Left and Vector to Bottom Right).
* The plane unit normal should be collinear with GCC [0,0,0].

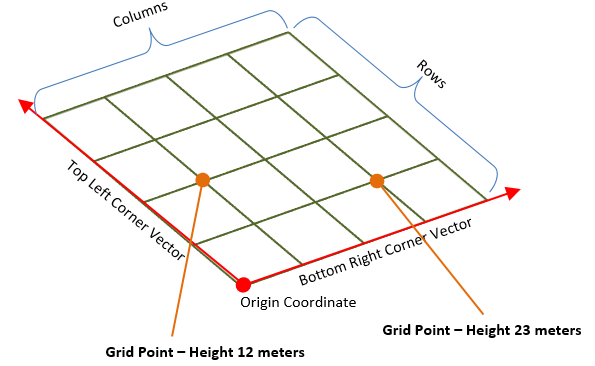


Figure ‑. Graphical representation of the Scaled Height Map. A height is listed for each Grid Point.

#### Receipt of the Scaled Height Map PDU

The Scaled Height Map PDU shall be received by a simulation application in response to a simulation event or occurrence that alters the terrain skin. Responses that span multiple PDUs shall have matching Object ID and Update Number fields, and PDU Number and PDU Total fields set appropriately.

The receiving application shall update its internal representation of the terrain surface in the area represented in the PDU according to the algorithm outlined in 5.11.2.4.4. No additional PDU is required to be issued after terrain updates are complete.

#### Scaled Height Map PDU

Information about terrain heights at regular intervals shall be communicated using a Scaled Height Map PDU. If a scaled height map message is too big to fit into one PDU due to network MTU size issues (or other parameters as defined by the simulation application or per exercise agreement), the PDU Number and PDU Total fields shall be used to indicate that the entire message spans multiple PDUs. In addition, the Object ID and Update Number fields will match. The Scaled Height Map PDU shall contain the following fields:

1. *PDU Header*. This field shall contain data common to all DIS PDUs. The PDU Header shall be represented by the PDU Header record (see 6.2.66).
2. *Object ID*. This field shall identify the entity or object associated with the instance of modified terrain defined by the Scaled Height Map. This field shall be represented by an Object Identifier record (see 6.2.63).
3. *Referenced Object ID*. This field shall identify the object which this object is associated. This field shall be represented by an Object Identifier record (see 6.2.63).
4. *Update Number*. This 16-bit unsigned integer shall identify the update number for this 2D Grid.
5. *Force ID*. This 8-bit enumeration (see [UID 6]) shall identify the force for which this terrain update applies.
6. *Object Type*. This field shall identify the type of object being updated. This field shall be represented by an Object Type record (see 6.2.64).
7. *Requester Simulation ID*. This field shall identify the simulation application requesting this terrain update. This field shall be represented by a Simulation Address record (see 6.2.80).
8. *Receiving Simulation ID*. This field shall identify the simulation application receiving this terrain update. This field shall be represented by a Simulation Address record (see 6.2.80).
9. *PDU Number*. This 16-bit unsigned integer shall identify the index of this PDU relative to the total number of the PDUs that make up the height changes of the specified 2D gridded terrain area.
10. *PDU Total*. This 16-bit unsigned integer shall identify the total number of PDUs that comprise the entire set of PDUs that describe the changes to the specified 2D gridded terrain area.
11. *Number of Rows*. This 16-bit unsigned integer shall identify the total number of rows for the 2D grid. This number will be the same for all associated PDUs that make up the complete grid transmission.
12. *Number of Columns*. This 16-bit unsigned integer shall identify the total number of columns for the 2D grid. This number will be the same for all associated PDUs that make up the complete grid transmission.
13. *Number of Points*. This 16-bit unsigned integer shall identify the total number of points being transmitted in this PDU. If only one PDU is required to transmit all the points in this grid, this value will be equal to the number of rows multiplied by the number of columns.
14. *Scale*. This 32-bit floating point shall identify the value that will be used to scale the elevation offsets.
15. *Origin Coordinate*. This field shall identify the location of the grid’s origin as the Bottom Left of the area. This field shall be represented by a World Coordinates record (see 6.2.98).
16. *Top Left Corner Vector*. This field shall identify a 3D vector from the Origin to the Top Left corner of the 2D grid. This value shall be represented by an Entity Coordinate vector (see 6.2.96).
17. *Bottom Right Corner Vector*. This field shall identify a 3D vector from the Origin to the Bottom Right corner of the 2D grid. This value shall be represented by an Entity Coordinate vector (see 6.2.96).
18. *Heights.* These 16-bit unsigned integers shall identify the collection of post definition heights in meters. Each post will define an offset from the plane, defined by the Origin and the Top Left and Bottom Right vectors, and along the unit normal from that plane. The grid posts shall be ordered in the record in “row major” order, with the first post being the Origin of the gridded area.
19. *Number of Height Map Information Records.* This 16-bit unsigned integer shall specify the number of Height Map Information Records.
20. *Height Map Information Records.* These fields shall contain one or more DT records that conform to the variable record format of a Standard Variable Specification record (see 6.2.83) and may contain other Standard Variable records.

The format of the Scaled Height Map PDU shall be as shown in Table 11‑6.

Table ‑. Scaled Height Map PDU

| Field Size  (bits) | Field Name | Data type |
| --- | --- | --- |
| 96 | PDU Header | Protocol Version - 8-bit enumeration |
| Exercise ID - 8-bit unsigned integer |
| PDU Type - 8-bit enumeration |
| Protocol Family - 8-bit enumeration |
| Time Stamp - 32-bit unsigned integer |
| Length - 16-bit unsigned integer |
| PDU Status - 8-bit record |
| Padding - 8-bits unused |
| 48 | Object ID | Site Number - 16-bit unsigned integer |
| Application Number - 16-bit unsigned integer |
| Reference Number - 16-bit unsigned integer |
| 48 | Referenced Object ID | Site Number - 16-bit unsigned integer |
| Application Number - 16-bit unsigned integer |
| Reference Number - 16-bit unsigned integer |
| 16 | Update Number | 16-bit unsigned integer |
| 8 | Force ID | 8-bit enumeration |
| 8 | Padding | 8 bits unused |
| 32 | Object Type | Domain – 8-bit enumeration |
| Kind – 8-bit enumeration |
| Category – 8-bit enumeration |
| Subcategory – 8-bit enumeration |
| 32 | Requester Simulation ID | Site Number - 16-bit unsigned integer |
| Application Number - 16-bit unsigned integer |
| 32 | Receiving Simulation ID | Site Number - 16-bit unsigned integer |
| Application Number - 16-bit unsigned integer |
| 16 | PDU Number | 16-bit unsigned integer |
| 16 | PDU Total | 16-bit unsigned integer |
| 16 | Number of Rows | 16-bit unsigned integer |
| 16 | Number of Columns | 16-bit unsigned integer |
| 16 | Number of Points (*H*) | 16-bit unsigned integer |
| 16 | Padding | 16 bits unused |
| 32 | Scale | 32-bit floating point |
| 192 | Origin Coordinate | *X*-component–64-bit floating point |
| *Y*-component–64-bit floating point |
| *Z*-component–64-bit floating point |
| 96 | Top Left Corner Vector | *x*-component–32-bit floating point |
| *y*-component–32-bit floating point |
| *z*-component–32-bit floating point |
| 96 | Bottom Right Corner Vector | *x*-component–32-bit floating point |
| *y*-component–32-bit floating point |
| *z*-component–32-bit floating point |
| 16 | Height *#1* | 16-bit unsigned integer |
| .  .  . | | |
| 16 | Heights *#H* | 16-bit unsigned integer |
| 8*P* | Padding | Padding to 48-bit boundary—*P* octets |
| 16 | Number of Height Map Information Records (*N*) | 16-bit unsigned integer |
| 48 + 8*K1* + 8*P1* | Height Map Information Record #1 | Record Type—32-bit enumeration |
| Record Length—16-bit unsigned integer  (6 + *K1* + *P1*) |
| Record-Specific fields—*K1* octets |
| Padding to 64-bit boundary—*P1* octets |
| .  .  . | | |
| 48 + 8*KN* + 8*PN* | Height Map Information Record #N | Record Type—32-bit enumeration |
| Record Length—16-bit unsigned integer  (6 + *KN* + *PN*) |
| Record-Specific fields—*KN* octets |
| Padding to 64-bit boundary—*PN* octets |
| Total Scaled Height Map PDU = 848 + 16*H* + 8P + 8  where  *H* is Number of Points and Height Values  *N* is the number of Height Map Information records  *Ki* is the length of the Record-Specific field in Height Map Information Records record #*i* in octets  *Pi* is the number of padding octets in Height Map Information Records record *i*, which is ⌈*(6+Ki)/8*⌉*8*-*(6+Ki)*  ⌈*x*⌉ is the largest integer < *x* + 1. | | |

### Soil Attribute Map PDU

#### Purpose

The Soil Attribute Map PDU communicates soil attributes inside a predefined, 3D rectangular area of the terrain. Each data item in the PDU represents an attribute of the soil, represented by an enumerated value for the particular attribute type.

#### Information contained in the Soil Attribute Map PDU

The Soil Attribute Map PDU shall contain the following information:

1. Standard PDU header data
2. Identification information for this terrain update
3. Data describing the area of terrain being updated
4. Data describing the soil attributes of the terrain throughout the area being updated
5. Data describing the values used to pack and unpack the data in the PDU

#### Issuance of the Soil Attribute Map PDU

The Soil Attribute Map PDU shall be issued by a simulation application in response to a simulation event or occurrence that alters the soil attributes of the terrain. Depending on the size of the terrain area being updated and the resolution of the attribute map, the update may be split into multiple PDUs. The PDUs will be associated via matching Object ID and Update Number fields, and the PDU Number and PDU Total fields.

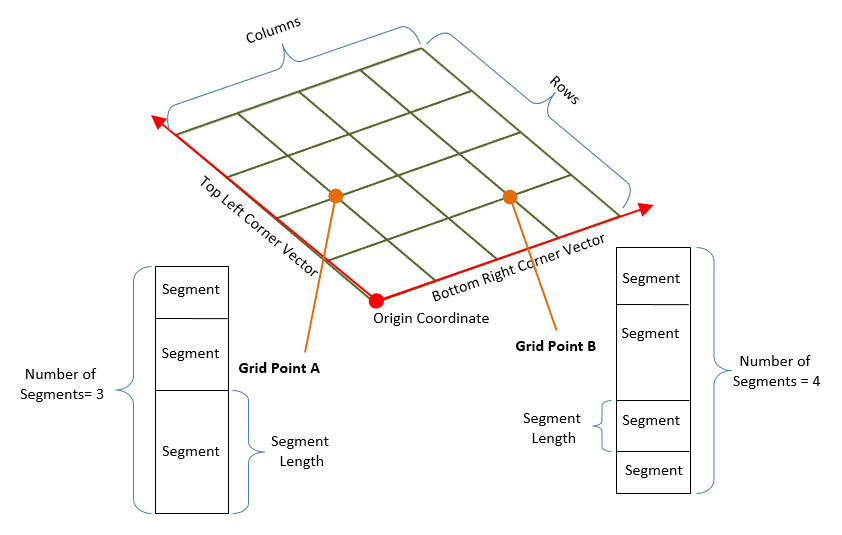


Figure ‑. Soil Attribute Map PDU Fields

This PDU represents a soil attribute rectangular grid. At each point within the grid, there are zero or more segments (up to 255) that define the soil composition attributes at that location. Each grid point may have a different number of segments. In the example above, Grid Point A has 3 segments and Grid Point B has 4 segments. The soil is described using a set of attributes (listed in the Attribute Type fields of the PDU). Each segment has an associated value for each Attribute Type.

#### Receipt of the Soil Attribute Map PDU

The Soil Attribute Map PDU shall be received by a simulation application in response to a simulation event or occurrence that alters the soil attributes of the terrain. Responses that span multiple PDUs shall have matching Object ID and Update Number fields, and PDU Number and PDU Total fields set appropriately.

The receiving application shall update its internal representation of the terrain soil attributes in the area represented in the PDU. No additional PDU is required to be issued after soil attribute updates are complete.

#### Soil Attribute Enumeration

These are the attributes that can be used to describe each segment of the soil grid. The desired Soil Attribute Enumerations are represented in the Attribute Type fields. Each selected Attribute Type will have a corresponding value represented in the Attribute Value fields*.* For example, a simulation that is only concerned with representing clay soil could use the Clay Mass and Clay Volume Fraction types for the Attribute Type fields. Note that the sum of all volume fractions must not exceed 1 for a given segment. If the sum of all volume fractions is less than 1, then the remaining value remaining value represents the void fraction (which can consist of air and water).

Table ‑. Soil Attribute Enumeration

| Value | Attribute Type | Attribute Description |
| --- | --- | --- |
| 0 | Clay Mass | Mass of the clay within the segment |
| 1 | Clay Volume Fraction | [0 to 1] Fraction of the segment that is clay |
| 2 | Silt Mass | Mass of the silt within the segment |
| 3 | Silt Volume Fraction | [0 to 1] Fraction of the segment that is silt |
| 4 | Sand Mass | Mass of the sand within the segment |
| 5 | Sand Volume Fraction | [0 to 1] Fraction of the segment that is sand |
| 6 | Gravel Mass | Mass of the gravel within the segment |
| 7 | Gravel Volume Fraction | [0 to 1] Fraction of the segment that is gravel |
| 8 | Water Mass | Mass of the water within the segment |
| 9 | Water Volume Fraction | [0 to 1] Percent of the segment that is water |
| 128 | Water Phase | [-1 to 1] Value of -1 is frozen water (ice). 0 is liquid water, and 1 is vapor. |

#### Example of Data within a Soil Attribute Map PDU



Figure ‑. Example of a Simple Grid (not to scale)

This example contains an oversimplified grid with only one square, and describes how to represent the soil composition data within this PDU. Not all fields are included in the example, but there are enough to help explain how to put your data in the correct format. None of these values are scaled in order to more clearly explain the fields. In practice, you should use the scaling formulas as provided in Section 7.10.8.

Table ‑. Values in the Soil Attribute Map PDU Based on Example

| PDU Field | PDU Value | Description |
| --- | --- | --- |
| Number of Rows (R) | 1 | Single square = 1 row |
| Number of Columns (C) | 1 | Single square = 1 column |
| Number of Points (P) | 4 | A point on each corner of the square |
| Number of Attribute Types (A) | 6 | Clay, Sand, and Gravel have been chosen, and each has a mass and volume, so A = 6. See the Attribute Types field. |
| … additional fields | | |
| Maximum Segment Mass (M) | 8 | The max segment mass in the provided example is 8 kg. |
| Maximum Segment Length (L) | 5 | The max segment length in the provided example is 5 meters. |
| … additional fields | | |
| Number of Segments (One field for each grid point - #P) | 2 | Grid Point A has 2 segments |
| 4 | Grid Point B has 4 segments |
| 2 | Grid Point C has 2 segments |
| 3 | Grid Point D has 3 segments |
| Segment Length (One field for each segment - #S) | 3 | Grid A, Segment 1 has length 3 meters |
| 3 | Grid A, Segment 2 has length 3 meters |
| 2 | Grid B, Segment 1 has length 2meters |
| 5 | Grid B, Segment 2 has length 5 meters |
| 2 | Grid B, Segment 3 has length 2 meters |
| 1.5 | Grid B, Segment 4 has length 1.5 meters |
| 4 | Grid C, Segment 1 has length 4 meters |
| 2 | Grid C, Segment 2 has length 2 meters |
| 2 | Grid D, Segment 1 has length 2 meters |
| 3 | Grid D, Segment 2 has length 3 meters |
| 5 | Grid D, Segment 3 has length 5 meters |
| Attribute Value (One field for each attribute value for each segment - #A \* S) | 3 | Grid A, Segment 1 Clay Mass in kg |
| 0.5 | Grid A, Segment 1 Clay Volume Fraction |
| 0 | Grid A, Segment 1 Sand Mass in kg |
| 0 | Grid A, Segment 1 Sand Volume Fraction |
| 5 | Grid A, Segment 1 Gravel Mass in kg |
| 0.5 | Grid A, Segment 1 Gravel Volume Fraction |
| … (continues with 6 attribute values for each segment) | |
| 0 | Grid D, Segment 3 Clay Mass in kg |
| 0 | Grid D, Segment 3 Clay Volume Fraction |
| 4 | Grid D, Segment 3 Sand Mass in kg |
| 0.6 | Grid D, Segment 3 Sand Volume Fraction |
| 8 | Grid D, Segment 3 Gravel Mass in kg |
| 0.4 | Grid D, Segment 3 Gravel Volume Fraction |
| Attribute Type (One enumeration value for each chosen attribute - #A) | 0 | Clay Mass |
| 1 | Clay Volume Fraction |
| 4 | Sand Mass |
| 5 | Sand Volume Fraction |
| 6 | Gravel Mass |
| 7 | Gravel Volume Fraction |

#### Soil Attribute Map PDU

Information about soil composition at regular intervals shall be communicated using a Soil Attribute Map PDU. See 5.11.2.5 for additional description and specific requirements on the use of the Soil Attribute Map PDU. If a soil attribute map message is too big to fit into one PDU due to network MTU size issues (or other parameters as defined by the simulation application or per exercise agreement), the PDU Number and PDU Total fields shall be used to indicate that the entire message spans multiple PDUs. In addition, the Object ID and Update Number fields will match.

Values within the Soil Attribute Map PDU are scaled to allow more data to fit within each PDU. The Maximum Segment Mass and Maximum Segment Length fields are used to scale the segment mass attribute values and the segment length values, respectively.

1. *PDU Header*. This field contains data common to all DIS PDUs. The PDU Header record shall represent this field (see 6.2.66).
2. *Object ID*. This field shall identify the entity or object associated with the instance of modified terrain defined by the Soil Attribute Map. This field shall be represented by an Object Identifier record (see 6.2.63).
3. *Referenced Object ID*. This field shall identify the object which this object is associated. This field shall be represented by an Object Identifier record (see 6.2.63)
4. *Update Number*. This 16-bit unsigned integer shall identify the update number for this 2D Grid.
5. *Force ID*. This 8-bit enumeration (see [UID 6]) shall identify the Force for which this terrain update applies.
6. *Object Type*. This field shall identify the type of object being updated. This field shall be represented by an Object Type record (see 6.2.64).
7. *Requestor Simulation ID*. This field shall identify the simulation application requesting this terrain update. This field shall be represented by a Simulation Address record (see 6.2.80).
8. *Receiving Simulation ID*. This field shall identify the simulation application receiving this terrain update. This field shall be represented by a Simulation Address record (see 6.2.80).
9. *PDU Number*. This 16-bit unsigned integer shall identify the index of this PDU relative to the total number of the PDUs that make up the soil attribute changes of the specified 2D gridded terrain area.
10. *PDU Total*. This 16-bit unsigned integer shall identify the total number of PDUs that comprise the entire set of PDUs that describe the changes to the specified 2D gridded terrain area.
11. *Number of Rows*. This 16-bit unsigned integer shall identify the total number of rows for the 2D grid. This number will be the same for all associated PDUs that make up the complete grid transmission.
12. *Number of Columns*. This 16-bit unsigned integer shall identify the total number of columns for the 2D grid. This number will be the same for all associated PDUs that make up the complete grid transmission.
13. *Number of Points*. This 16-bit unsigned integer shall identify the total number of points stored in this PDU. If all data for this grid fits within one PDU (PDU Total = 1), this value will equal Number of Rows \* Number of Columns, otherwise it will be the Number of Points that are contained within this PDU.
14. *Number of Attribute Types*. This 8-bit unsigned integer shall identify the total number of attribute types in this PDU. This value is the number of Attribute Type fields.
15. *Maximum Segment Mass*. This 32-bit unsigned integer shall identify the maximum total mass of a segment within this PDU. This value is used to scale the attribute values to be more efficient with space within the PDU.
16. *Maximum Segment Length*. This 32-bit unsigned integer shall identify the maximum length of a segment within this PDU. This value is used to scale the segment length values to be more efficient with space within the PDU.
17. *Origin Coordinate*. This field shall identify the GCC location of the grid’s Origin (for clarity, consider this the Bottom Left corner).
18. *Top Left Corner Vector*. This field shall identify a 3D vector from the Origin (Bottom Left) to the Top Left corner of the 2D grid. This value shall be represented by an Entity Coordinate vector (see 6.2.96).
19. *Bottom Right Corner Vector*. This field shall identify a 3D vector from the Origin (Bottom Left) to the Bottom Right corner of the 2D grid. This value shall be represented by an Entity Coordinate vector (see 6.2.96).
20. *Number of Segments.* This series of 8-bit unsigned integers contains a number of segments for each grid post. Each grid post may have a different number of segments. It may contain a byte of padding, determined by the following formula:

#padding\_bytes = CEIL ((Number of Points / 2) \* 2 – Number of Points)

1. *Segment Lengths.* This series of 16-bit unsigned integers represents the length (in meters) of each segment in the PDU. Each point has a different number of segments, and each of these has a length (could also be considered a depth, where the sum of the depths of each segment at a point is the post depth at that point). These values are scaled to condense their potential bit usage according to the following formula:

packed\_segment\_length = FLOOR (((segment length (meters) / Maximum Segment Length) \* 65535) + 0.5)

1. *Attribute Value.* This series of 8-bit unsigned integers represents the chosen Attribute Type’s value. There is a value for each attribute, for each segment. The set of chosen attributes is defined in the next field (*Attribute Type*), and is a subset of the Soil Attribute Enumeration (See **Error! Reference source not found.**). The units for Mass are in kg. The values associated with a Mass (See Table XXX) are scaled to condense their potential bit usage according to the following formula:

packed\_segment\_mass = FLOOR (((mass / Maximum Segment Mass) \* 255) + 0.5)

Volume fractions are packed according to the following formula:

packed\_segment\_volume\_fraction = FLOOR ((volume fraction \* 255) + 0.5)

1. *Attribute Type.* This series of 8-bit enumerations represents the selected Attribute Types utilized in the PDU. The set of Attribute Types chosen for each Soil Update is a subset of the Soil Attribute Enumeration set. For each Attribute Type chosen from the Soil Attribute Enumeration (See **Error! Reference source not found.**) there is one of these fields.
2. *Number of Soil Attribute Information Records.* This 16-bit unsigned integer shall specify the number of Soil Attribute Map Information Records.
3. *Soil Attribute Information Records.* These fields shall contain one or more DT records that conform to the variable record format of a Standard Variable Specification record (see 6.2.83) and may contain other Standard Variable records.

The format of the Soil Attribute Map PDU shall be as shown in Table 11‑9.

Table ‑. Soil Attribute Map PDU

| Field Size  (bits) | Field name | Data type |
| --- | --- | --- |
| 96 | PDU Header | Protocol Version - 8-bit enumeration |
| Exercise ID - 8-bit unsigned integer |
| PDU Type - 8-bit enumeration |
| Protocol Family - 8-bit enumeration |
| Time Stamp - 32-bit unsigned integer |
| Length - 16-bit unsigned integer |
| PDU Status - 8-bit record |
| Padding - 8-bits unused |
| 48 | Object ID | Site Number - 16-bit unsigned integer |
| Application Number - 16-bit unsigned integer |
| Reference Number - 16-bit unsigned integer |
| 48 | Referenced Object ID | Site Number - 16-bit unsigned integer |
| Application Number - 16-bit unsigned integer |
| Reference Number - 16-bit unsigned integer |
| 16 | Update Number | 16-bit unsigned integer |
| 8 | Force ID | 8-bit enumeration |
| 8 | Padding |  |
| 32 | Object Type | Domain – 8-bit enumeration |
| Entity Kind – 8-bit enumeration |
| Category – 8-bit enumeration |
| Sub-Category – 8-bit enumeration |
| 32 | Requestor ID | Site Number - 16-bit unsigned integer |
| Application Number - 16-bit unsigned integer |
| 32 | Referenced Object ID | Site Number - 16-bit unsigned integer |
| Application Number - 16-bit unsigned integer |
| 16 | PDU Number | 16-bit unsigned integer |
| 16 | PDU Total | 16-bit unsigned integer |
| 16 | Number of Rows (R) | 16-bit unsigned integer |
| 16 | Number of Columns (C) | 16-bit unsigned integer |
| 16 | Number of Points (P) | 16-bit unsigned integer |
| 8 | Number of Attribute Types (A) | 8-bit unsigned integer |
| 8 | Padding |  |
| 32 | Maximum Segment Mass (M) | 32-bit unsigned integer |
| 32 | Maximum Segment Length (L) | 32-bit unsigned integer |
| 32 | Padding |  |
| 192 | Origin Coordinate | *X*-component–64-bit floating point |
| *Y*-component–64-bit floating point |
| *Z*-component–64-bit floating point |
| 96 | Top Left Corner Vector | *x*-component–32-bit floating point |
| *y*-component–32-bit floating point |
| *z*-component–32-bit floating point |
| 96 | Bottom Right Corner Vector | *x*-component–32-bit floating point |
| *y*-component–32-bit floating point |
| *z*-component–32-bit floating point |
| 8 | Number of Segments *#1* | 8-bit unsigned integer |
| … | | |
| 8 | Number of Segments *#P* | 8-bit unsigned integer |
| 16 | Segment Length *#1* | 16-bit unsigned integer |
| … | | |
| 16 | Segment Length *#S* | 16-bit unsigned integer |
| 8 | Attribute Value *#1* | 8-bit unsigned integer |
| … | | |
| 8 | Attribute Value *#(S \* A)* | 8-bit unsigned integer |
| 8 | Attribute Type *#1* | 8-bit enumeration |
| … | | |
| 8 | Attribute Type *#A* | 8-bit enumeration |
| 0-48 | Padding to 48-bit boundary | P*i* bits unused |
| 16 | Number of Soil Attribute Information Records | 16-bit unsigned integer |
| 48 + 8*K1* + 8*P1* | Soil Attribute Information Records #1 | Record Type—32-bit enumeration |
| Record Length—16-bit unsigned integer  (6 + *K1* + *P1*) |
| Record-Specific fields—*K1* octets |
| Padding to 64-bit boundary—*P1* octets |
| .  .  . | | |
| 48 + 8*KN* + 8*PN* | Soil Attribute Information Records #N | Record Type—32-bit enumeration |
| Record Length—16-bit unsigned integer  (6 + *KN* + *PN*) |
| Record-Specific fields—*KN* octets |
| Padding to 64-bit boundary—*PN* octets |
| Soil Attribute Map PDU = 912 + 8P + (16 + 8A)\* + 8A + 8  where  *P* is the number of Points  *A* is the number of Attribute Types  *S* is the number of Segments  *N* is the number of Soil Attribute Information records  *Ki* is the length of the Record-Specific field in Soil Attribute Information Records record #*i* in octets  *Pi* is the number of padding octets in Soil Attribute Information Records record #*i*, which is ⌈*(6+Ki)/8*⌉-*(6+Ki)*  ⌈*x*⌉ is the largest integer < *x* + 1. | | |

# DISTRIBUTED EMISSION REGENERATION (DER)

## Electromagnetic Emission

VDIS adopts the Electromagnetic Emission PDU usage defined in DIS.

### Jamming Technique

The Jamming Technique enumerations currently supported by VDIS are given in the following table. These are consistent with and a subset of the table in SISO-REF-010-00v20-0.

Table ‑. Jamming Technique Enumerations

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| KIND | CAT | SCAT | SPEC | Name |
| 0 | 0 | 0 | 0 | Other |
| 1 | 0 | 0 | 0 | Noise |
| 2 | 0 | 0 | 0 | Deception |

### Electromagnetic Emission PDU Issuance Rules

VDIS only recommends transmitting the Electromagnetic Emission PDU using the complete-entity issuance method. To support compatibility, however, VDIS simulation applications shall implement the Beam Status field.

### Electromagnetic Emission PDU Receipt Rules

It’s critical for VDIS compliant applications to support the Beam Status field so that they will not be affected by other simulation applications using the other issuance methods.

## Designator

VDIS adopts the Designator PDU usage defined in SISO PCR 130:

1. Clarified the PDU usage to include designation functions not strictly related to weapons engagement, such as laser range finding, beaming, and pointing.
2. Clarified the notion of a “designated entity” as being either an entity or an object. For this, “Designated Entity ID” is now “Designated Object ID” and “Designator Spot with respect to Designated Entity” is now “Designator Spot with respect to Designated Object.”
3. Changed the definition of Designator Code from an enumeration to an unsigned integer to allow usage of real world designation codes.
4. Added an unsigned integer field to the PDU in the 24-bit padding to indicate the Flash Rate of the laser designator.
5. Added a Designator System Number field to the PDU in the 24-bit padding to differentiate between possible multiple laser designation systems on the same entity. Each laser designator system shall be assigned a unique number to be included in this field. Systems shall be numbered sequentially starting from zero. Once established for an exercise, the Designator System Number for each designator system shall not be changed for that exercise regardless of whether it is active or inactive.
6. Added a Laser Function field to the PDU in the 24-bit padding to identify the mode of operation for the “designating” laser (see below for an enumeration).
7. Changed “Code Name” to “Designator System Name” to clarify its usage. Reduced the Designator System Name to an 8-bit enumeration.
8. Added “Designator Spot Type”, an 8-bit enumeration, in the padding just created by the reduction in size of the Designator System Name. This field identifies the type of lased spot.
9. Changed the “Entity Linear Acceleration” to “Designator Beam Origin Offset”, because the origin is far more useful feature than the dead reckoning parameters. Note that Entity Linear Acceleration is unusable without velocity, but even switching acceleration for velocity wouldn’t be as useful as the beam origin. This does not break backward compatibility because acceleration could not have been used previously.
10. VDIS provides the following dead reckoning clarification. If an object is specified as being tracked, then the dead reckoning is relative to the body coordinates of the tracked object using the Designator Spot with Respect to Designated Object field (i.e., DRM 9 FVB). If no object is being tracked, then the dead reckoning is based on the Designator Spot Location (i.e., DRM 5 FVW).
11. Enumerations for new fields (Designator Spot Type, Laser Function) as well as revisions to existing enumerations to support the Designator PDU (Designator System Name, Sensor/Emitter entity types).
12. Extended the PDU definition to include three Attribute records for an Attribute PDU associated to the Designator PDU:
    1. Designator Beam Characteristics Attribute record
    2. Designator Dead Reckoning Parameters Attribute record
    3. Designator Objects Attribute record

Table ‑. Designator PDU

| Field Size  (bits) | Designator PDU Fields | |
| --- | --- | --- |
| 96 | PDU Header | Protocol Version – 8-bit enumeration |
| Exercise ID – 8-bit unsigned integer |
| PDU Type – 8-bit enumeration |
| Protocol Family – 8-bit enumeration |
| Timestamp – 32-bit unsigned integer |
| Length – 16-bit unsigned integer |
| PDU Status – 8-bit enumeration |
| Padding – 8 bits unused |
| 48 | Designating Entity ID | Site Number – 16-bit unsigned integer |
| Application Number – 16-bit unsigned integer |
| Entity Number – 16-bit unsigned integer |
| 8 | Designator Spot Type | 8-bit enumeration |
| 8 | Designator System Name | 8-bit enumeration |
| 48 | Designated Object ID | Site Number – 16-bit unsigned integer |
| Application Number – 16-bit unsigned integer |
| Entity/Object Number – 16-bit unsigned integer |
| 16 | Designator Code | 16-bit unsigned integer |
| 32 | Designator Power | 32-bit floating point |
| 32 | Designator Wavelength | 32-bit floating point |
| 96 | Designator Spot with respect to Designated Object | x-component – 32-bit floating point |
| y-component – 32-bit floating point |
| z-component – 32-bit floating point |
| 192 | Designator Spot Location | x-component – 64-bit floating point |
| y-component – 64-bit floating point |
| z-component – 64-bit floating point |
| 8 | Dead Reckoning Algorithm | 8-bit enumeration |
| 8 | Flash Rate | 8-bit unsigned integer |
| 8 | Designator System Number | 8-bit unsigned integer |
| 8 | Laser Function | 8-bit enumeration |
| 96 | Designator Beam Origin Offset | x-component – 32-bit floating point |
| y-component – 32-bit floating point |
| z-component – 32-bit floating point |
| Total Designator PDU size = 704 bits | | |

### Designator PDU Enumerations

The Laser Function enumeration follows:

Table ‑. Laser Function Enumeration

| Value | Description |
| --- | --- |
| 0 | Designating |
| 1 | Ranging |
| 2 | IR Designating |
| 3 | Beam Riding |
| 4 | Laser Pointing |
| 5 | IR Marking |
| 6 | NVG Marking |

The Designator Spot Type enumeration follows:

Table ‑. Designator Spot Type Enumeration

| Value | Description |
| --- | --- |
| 0 | Not specified |
| 1 | Terrain – terrain model |
| 2 | Terrain – no terrain model (zero-elevation) |
| 3 | Water surface |
| 4 | Foliage or other natural object attached to terrain |
| 5 | Man-made object attached to terrain |
| 6 | Clouds |
| 7 | Obscurant (smoke, etc.) |
| 8 | Non-incident (e.g., clear sky) |

The revised Designator System Name enumeration follows:

Table ‑. Designator System Name Enumeration

| Value | Description |
| --- | --- |
| 0 | No Statement |
| 1 | AN/AAQ-16: FLIR Imaging System |
| 2 | AN/AAQ-22A: SAFIRE-LRF (USMC UH1N, MH60G) |
| 3 | AN/AAQ-22B: SAFIRE-LP (USAF) |
| 4 | AN/AAQ-22C: Star SAFIRE-I LRF (USMC) |
| 5 | AN/AAQ-22D: BRITE Star (USMC) |
| 6 | AN/AAQ-28: LITENING Airborne Targeting and Navigation POD |
| 7 | AN/AAQ-7 Laser Range Finder (AC-130A/H) |
| 8 | AN/AAS-35(V): Pave Penny Airborne Ground Attack Pod |
| 9 | AN/AAS-37 Laser Range Finder and Designator (OV-10 and OV-1D) |
| 10 | AN/AAS-38 NITE HAWK FLIR and Laser Designator Pod (F/A-18) |
| 11 | AN/AAS-44 IR Laser Detecting/Ranging/Tracking Set (SH-60B) |
| 12 | AN/AAS-49 IR Laser Detecting/Ranging/Tracking Set (derivative of AN/AAS-44) |
| 13 | AN/AAS-51 IR Laser Detecting/Ranging/Tracking Set (derivative of AN/AAS-44 used in helicopters) |
| 14 | AN/AAS-53 Electro Optical/IR/Laser sensor turret for Army ARH (AN/AAS-52 in MQ-1) |
| 15 | AN/GVS-5: Hand Held LRF |
| 16 | AN/TVQ-2: Ground Vehicle Laser Locator Designator (GVLLD) |
| 17 | DAK-1: Russian LRF Sage Gloss (artillery weapons) |
| 18 | DHY 307: Ground Laser Designator |
| 19 | Semi Automatic Command Line of Sight (SACLOS) - Wired / Beam Rider / Laser Designator |
| 20 | TPDK-1: Russian LRF |
| 21 | 1D15: Russian Laser Target Designator |
| 22 | 1D20: Russian Laser Target Designator |
| 23 | 1D22: Russian Laser Target Designator |
| 24 | IZLID 428P-W |
| 25 | AIM-1/SLX Super Long Range IR Aiming Laser (Helicopter mounted) |
| 26 | AIM-1/SLR Super Long Range IR Aiming Laser |
| 27 | AN/PEQ-2A IR Aiming Laser |
| 28 | ACP-2A Air Commander Pointer IR Aiming Laser |

The location of the spot referenced in the Designator PDU must be ground truth data. If dead reckoning is required by the simulation application (or per exercise agreement), then the simulation application modeling the designated spot shall dead reckon the spot using the dead reckoning parameters published in a coupled Attribute PDU so that receivers of the Designator PDU can also dead reckon the movement of the spot.

Whenever a simulation application designates a target with a simulated laser designator or designates an entity or spot in the world for laser range finding, laser pointing, or laser beaming the simulation application issues the result in the Designator PDU. Simulation applications that simulate sensors that can detect laser use are made aware of the lasing activity.

### Flashing Lasers and Power Modes

Simulation applications desiring to implement or representing flashing, or pulsing, lasers, shall utilize the Flash Rate field in the Designator PDU. This field identifies the flash rate in seconds (Hz) of the designator. The Flash Rate shall contain a value of zero if the designator system is not flashing (zero does not indicate that the system is off). If the system is flashing, then the Flash Rate field shall contain a valid flash rate in hertz. If the Designator Power field indicates that the power is off (zero), then any non-zero flash rate value shall be reset to zero.

To signify differing power modes, such as high or low, the simulation application shall set the Designator Power to the appropriate value. If a specific mode is required to be identified, then it is recommended that the Designator Beam Characteristics Attribute Record be modified to provide an 8-bit enumeration.

## Supplemental Emission / Entity State (SEES)

The Supplemental Emission/Entity State (SEES) PDU provides the capability to communicate supplemental emissions and entity state attributes. VDIS has no recommendation for usage of this PDU. It was considered for usage in high fidelity thermal simulation, but the single 16-bit infrared index (basically an enumeration) is not well-defined and otherwise insufficient. The number of propulsion systems and nozzles is helpful, but may not correlate directly with thermal signatures.

## Identification, Friend, or Foe (IFF)

VDIS will adopt the IFF PDU as proposed in DIS, but currently only recommends the usage of layer 1 to support the use cases for Mode 4 operations.

VARIABLE RECORDS

This appendix defines all variable and fixed records that are directly supported in VDIS but are not otherwise defined in this main specification document.

Please refer to the external file titled VDIS\_Spec\_Appendix\_A.doc for further information.

ENTITY SPECIFIC INFORMATION

This appendix defines all entity specific information as needed for interoperability. This information will not identify those items that are not available at the distribution level of this document.

HMMWV: 1.1.225.6.1.X.X

HMMWV variants may use several types of weapons. VDIS identifies the weapon type for particular entities within the articulated part with the documented mapping given below. This mapping is currently valid only with the following HMMWV variants:

* M998 (1.1.225.6.1.1.X)
* M1026 (1.1.225.6.1.8.X)
* M707 (1.1.225.6.1.20.X)
* M1114 (1.1.225.6.1.21.X)
* M1151 (1.1.225.6.1.32.X)

Other variants of the HMMWV are not yet supported by this specification. These weapons are mutually exclusive, i.e., the system doesn’t support multiple primary guns or a primary gun and a primary launcher. If multiple weapons are available and supported, it is recommended that all other weapons use the secondary classifications. Note that the current requirement is to support these guns with a changing elevation and possibly elevation rate. The combined articulation is also identified below.

* M2 – Primary Gun Number 1 (4416)
  + Elevation: +13 (4429)
  + Elevation Rate: +14 (4430)
* M240 – Primary Gun Number 2 (4448)
  + Elevation: +13 (4461)
  + Elevation Rate: +14 (4462)
* M249 – Primary Gun Number 3 (4480)
  + Elevation: +13 (4493)
  + Elevation Rate: +14 (4494)
* MK19 Grenade Launcher – Primary Launcher 1 (4736)
  + Elevation: +13 (4749)
  + Elevation Rate: +14 (4750)

The turret for each shall use Primary Turret Number 1 with an azimuth and azimuth rate articulation as shown below.

* Primary Turret Number 1 (4096)
  + Azimuth: +11 (4107)
  + Azimuth Rate: +12 (4108)

HEMTT: 1.1.225.7.19.X.X

HEMTT variants may use several types of weapons. VDIS identifies the weapon type for particular entities within the articulated part with the documented mapping given below. This mapping is currently valid only with the following HEMTT variants:

* M977 Cargo (1.1.225.7.19.1.X)
* M978 Fuel-servicing (1.1.225.7.19.2.X)

Other variants of the HEMTT are not yet supported by this specification. These weapons are mutually exclusive, i.e., the system doesn’t support multiple primary guns or a primary gun and a primary launcher. If multiple weapons are available and supported, it is recommended that all other weapons use the secondary classifications. Note that the current requirement is to support these guns with a changing elevation and possibly elevation rate. The combined articulation is also identified below.

* M2 – Primary Gun Number 1 (4416)
  + Elevation: +13 (4429)
  + Elevation Rate: +14 (4430)
* M240 – Primary Gun Number 2 (4448)
  + Elevation: +13 (4461)
  + Elevation Rate: +14 (4462)
* M249 – Primary Gun Number 3 (4480)
  + Elevation: +13 (4493)
  + Elevation Rate: +14 (4494)
* MK19 Grenade Launcher – Primary Launcher 1 (4736)
  + Elevation: +13 (4749)
  + Elevation Rate: +14 (4750)

The turret for each shall use Primary Turret Number 1 with an azimuth and azimuth rate articulation as shown below.

* Primary Turret Number 1 (4096)
  + Azimuth: +11 (4107)
  + Azimuth Rate: +12 (4108)

M1200 Armored Knight: 1.1.225.2.40.X.X

The M1200 Armored Knight may use several types of weapons. VDIS identifies the weapon type for particular entities within the articulated part with the documented mapping given below.

These weapons are mutually exclusive, i.e., the system doesn’t support multiple primary guns or a primary gun and a primary launcher. If multiple weapons are available and supported, it is recommended that all other weapons use the secondary classifications. Note that the current requirement is to support these guns with a changing elevation and possibly elevation rate. The combined articulation is also identified below.

* M2 – Primary Gun Number 1 (4416)
  + Elevation: +13 (4429)
  + Elevation Rate: +14 (4430)
* M240 – Primary Gun Number 2 (4448)
  + Elevation: +13 (4461)
  + Elevation Rate: +14 (4462)
* M249 – Primary Gun Number 3 (4480)
  + Elevation: +13 (4493)
  + Elevation Rate: +14 (4494)
* MK19 Grenade Launcher – Primary Launcher 1 (4736)
  + Elevation: +13 (4749)
  + Elevation Rate: +14 (4750)

The turret for each shall use Primary Turret Number 1 with an azimuth and azimuth rate articulation as shown below.

* Primary Turret Number 1 (4096)
  + Azimuth: +11 (4107)
  + Azimuth Rate: +12 (4108)

Cell Phone User

Several life form entity types may have and therefore use cell phones (e.g., Afghan civilian, cell phone, 3.1.1.3.0.0.1). VDIS identifies this use case and clarifies the usage of the Life Form Appearance record, specifically regarding bits 24-25 and 26-27, the Weapon/Implement fields. For the Weapon/Implement 1 field (24-25), set as follows:

* 0 – N/A
* 1 – Cell phone held at the side (stowed)
* 2 – Cell phone in hand, dialing (active)
* 3 – Cell phone held to ear (in use)

For the Weapon/Implement 2 field (26-27), set to Not Present (0).

M60A1 AVLB

The M60A1 (1.1.225.3.4.2.0) is an Armored Vehicle Launched Bridge that articulates a bridge and then detaches it for deployment. The VDIS mapping of the articulated parts are now reflected in IEEE Std 1278.1-2012.

M1 Abrams: 1.1.225.1.1.X.X

The M1A1 and M1A1D have the following parts:

* Main turret: Primary Turret Number 1 (4096)
  + Azimuth: +11 (4107)
  + Azimuth Rate: +12 (4108)
* Main gun: Primary Gun Number 1 (4416) (attached to main turret)
  + Elevation: +13 (4429)
  + Elevation Rate: +14 (4430)
* Cupola: Secondary Turret Number 1 (5696) (attached to main turret)
  + Azimuth: +11 (5707)
  + Azimuth Rate: +12 (5708)

The M1A1 SA and M1A1 FEP have the following parts:

* Main turret: Primary Turret Number 1 (4096)
  + Azimuth: +11 (4107)
  + Azimuth Rate: +12 (4108)
* Main gun: Primary Gun Number 1 (4416) (attached to main turret)
  + Elevation: +13 (4429)
  + Elevation Rate: +14 (4430)
* SCWS Azimuth: Secondary Turret Number 1 (5696) (attached to main turret)
  + Azimuth: +11 (5707)
  + Azimuth Rate: +12 (5708)
* 50 Cal: Secondary Gun Number 1 (6016) (attached to SCWS)
  + Elevation: +13 (6029)
  + Elevation Rate: +14 (6030)

The M1A2 SEP V2 has the following part:

* Main turret: Primary Turret Number 1 (4096)
  + Azimuth: +11 (4107)
  + Azimuth Rate: +12 (4108)
* Main gun: Primary Gun Number 1 (4416) (attached to main turret)
  + Elevation: +13 (4429)
  + Elevation Rate: +14 (4430)
* CROWS II Azimuth: Secondary Turret Number 1 (5696) (attached to main turret)
  + Azimuth: +11 (5707)
  + Azimuth Rate: +12 (5708)
* 50 Cal: Secondary Gun Number 1 (6016) (attached to CROWS II)
  + Elevation: +13 (6029)
  + Elevation Rate: +14 (6030)

KEY TERMS AND ACRONYMS

This appendix identifies the acronyms used or referenced in the VDIS specification.

|  |  |
| --- | --- |
| AAR | After Action Review |
| AFATDS | Advanced Field Artillery Targeting and Direction System |
| AMSAA | Army Materiel System Analysis Activity |
| AVCATT-A | Aviation Combined Arms Tactical Trainer-Aviation Reconfigurable Manned Simulator |
| BCIS | Battlefield Combat Identification System |
| BLUFOR | Blue Force |
| BMC | Battlemaster Control |
| C2 | Command and Control |
| C4I | Command, Control, Communications, Computers and Information |
| C4ISR | Command, Control, Communications, Computers, Intelligence, Surveillance, and Reconnaissance |
| CBRNE | Chemical, Biological, Radiological, Nuclear and High Explosive |
| CCTT | Close Combat Tactical Trainer |
| CDT | Common Driver Trainer |
| CDT/SV | Common Driver Trainer/Stryker Variant |
| CFFT | Call For Fire Trainer |
| CGF | Computer Generated Forces |
| COTS | Commercial Off The Shelf |
| CTIA | Common Training Information Architecture |
| CVEM | Common Virtual Environment Management |
| DI | Dismounted Infantry |
| DIS | Distributed Interactive Simulation |
| DOD | Department of Defense |
| DOF | Degrees of Freedom |
| ECCM | Electronic Counter Counter Measures |
| ECM | Electronic Counter Measures |
| ERC | Environment Run-time Component |
| FBCB2 | Force XXI Battle Command Battalion/Brigade and Below |
| FOV | Field of View |
| GCC | Geocentric Coordinates |
| GCS | Global Coordinate System |
| GDC | Geodetic Coordinates |
| GOTS | Government Off the Shelf |
| GUI | Graphical User Interface |
| HLA | High Level Architecture |
| HMD | Helmet Mounted Display |
| HMMWV | High Mobility Multipurpose Wheeled Vehicle |
| IED | Improvised Explosive Device |
| IG | Image Generator |
| JCATS | Joint Conflict and Tactical Simulation |
| JITC | Joint Interoperability Test Command |
| JRTC | Joint Readiness Training Center |
| LAN | Local Area Network |
| LHN | Long Haul Networking |
| LOD | Level of Detail |
| LVC | Live, Virtual, Constructive |
| LVC-IA | Live, Virtual, Constructive – Integrated Architecture |
| M&S | Modeling & Simulation |
| MCC | Master Control Console |
| MM | Manned Module |
| MOOTW | Military Operations Other Than War |
| MOUT | Military Operations in Urban Terrain |
| MSDL | Military Scenario Development Language |
| NATO | North Atlantic Treaty Organization |
| NBC | Nuclear, Biological and Chemical |
| NCBR | Nuclear, Chemical, Biological and Radiological |
| NTC | National Training Center |
| OneSAF | One Semi-Automated Forces |
| OneTESS | One-Tactical Engagement Simulation System |
| OPFOR | Opposing Force |
| OTW | Out The Window |
| PCR | Program Change Request |
| PEO STRI | Program Executive Office for Simulation Training and Instrumentation |
| PM | Program Manager |
| PM CATT | Project Manager, Combined Arms Tactical Trainer |
| PVD | Plan View Display |
| QoS | Quality of Service |
| RWA | Rotary-Wing Aircraft |
| SA | Situational Awareness |
| SAF | Semi-Automated Forces |
| SCIPUFF | Second Order Closure Integrated Puff |
| SEDRIS | Synthetic Environment Data Representation and Interchange Specification |
| SINCGARS | Single Channel Ground and Airborne Radio System |
| SISO | Simulation Interoperability Standards Organization |
| SNE | Synthetic Natural Environment |
| TADSS | Training Aids, Devices, Simulators and Simulations |
| TBD | To Be Determined |
| TENA | Test and Training Enabling Architecture |
| TOC | Tactical Operation Center |
| TOC ABCS | Tactical Observations Center Army Battle Command Systems |
| TCM | TRADOC Capability Management |
| TRADOC | Training and Doctrine Command |
| UAS | Unmanned Aerial System |
| UAV | Unmanned Aerial Vehicle |
| UHRB | Ultra-High Resolution Building |
| UKCATT | United Kingdom Combined Arms Tactical Trainer |
| USGS | U.S. Geological Survey |
| USMC | United States Marine Corps |
| UTM | Universal Transverse Mercator |
| VCCT | Virtual Convoy Combat Trainer |
| VPN | Virtual Private Network |
| WAN | Wide Area Network |
| WARSIM | Warfighters’ Simulation |
| XML | Extensible Markup Language |

1. <http://en.wikipedia.org/wiki/Oklahoma_City_bombing> [↑](#footnote-ref-1)